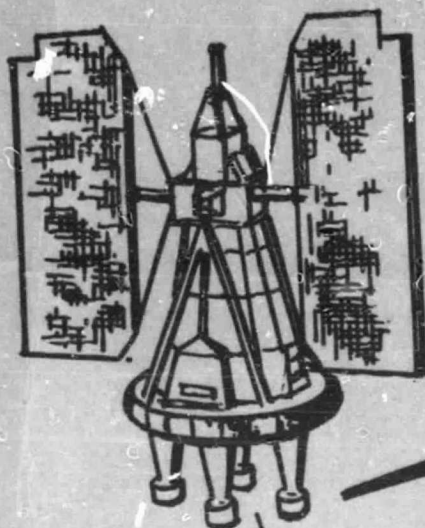


N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE AS MUCH
INFORMATION AS POSSIBLE

Onboard Utilization of Ground Control Points for Image Correction Final Report



(NASA-CR-166732) ONBOARD UTILIZATION OF
GROUND CONTROL POINTS FOR IMAGE CORRECTION.
VOLUME 2: ANALYSIS AND SIMULATION RESULTS
Final Report (Martin Marietta Corp.) 183 p
HC A09/AF A01

N82-10477

Unclass
39418
MARTIN MARIETTA

Volume II

Analysis and
Simulation
Results

May 1981

**ONBOARD UTILIZATION OF
GROUND CONTROL POINTS
FOR IMAGE CORRECTION
FINAL REPORT**

**MARTIN MARIETTA
DENVER AEROSPACE
Denver, Colorado 80201**

CONTENTS

	<u>Page</u>
I. INTRODUCTION	I-1 thru I-7
II. ONBOARD IMAGE CORRECTION SYSTEM OVERVIEW	II-1
A. Registration Processor	II-2
B. Navigation	II-17
C. Pointing Mount Controller and Linear Array	II-18 thru II-20
III. GCP NAVIGATION SYSTEM SIMULATION	III-1
A. Simulated GCP Measurement	III-1
B. Extracted GCP Measurements	III-30 thru III-32
IV. GCP ANALYSIS	IV-1
A. Error Budget	IV-1
B. Parametric Studies	IV-3
C. Overall Conclusion	IV-8
D. Recommendations	IV-8
E. Plots	IV-9 thru IV-110
V. IMAGE CORRECTION ACCURACY MEASUREMENT	V-1 thru V-4
VI. CONCLUSION AND RECOMMENDATIONS	VI-1
A.	VI-1
B. Recommendations	VI-3
VII. REFERENCES	VII-1

FIGURE

	<u>Page</u>
II-1 Viewing Perspective Distortions	I-4
II-1 Onboard Image Correction System Block Diagram	II-1
II-2 Functional Flow of Registration Processor	II-3
II-3 Organization of the GCP Data Base	II-3
II-4 Determination of Sensor Boresight Surface Position	II-4
II-5 Explanation of Parameters in Count Table	II-6
II-6 Search Area Algorithm	II-8
II-7 Subimage Definition	II-9
II-8 Registration of Sensor Data via GCP Detection	II-10
II-9 99% Confidence Polygons for Feature Classification	II-12
II-10 Parameters Involved in Data Selection Process	II-13
II-11 Correlator Design	II-16
II-12 Navigation Processor Functional Description	II-18
II-13 Curved Focal Plans Array.	II-20
III-1 Landmark Tracker Geometry - Sighting Plan	III-4
III-2 Landmark Tracker Geometry - General	III-4
III-3 Star Tracker Measurement Model Development	III-8
III-4 Overview of Measurement Models	III-12
III-5 Software Algorithm	III-15
III-6 Correlation - Scroll Program Structure	III-32
V-1 Reference Grid Pattern	V-1
V-2 GCP Selection	V-2
V-3 Registration Vector	V-3
V-4 Distortion Measurement Program	V-3 and V-4

TABLE

	<u>Page</u>
IV-1 Attitude Reference Unit Error Coefficients	IV-2
IV-2 Star Tracker Error Coefficients	IV-2
IV-3 GPS Error Coefficients	IV-2
IV-4 Landmark Tracker Error Coefficients	IV-3
IV-5 Cases Studied	IV-4
IV-6 Results of GCP Update Interval Sensitivity	IV-5
IV-6 Results of GPS Update Interval Sensitivity	IV-6

I. INTRODUCTION

Several current and anticipated trends threaten to limit the usefulness of future NASA remote sensing missions if we continue to employ current data handling methods. Forecasts of instrumentation trends show an increase in sensor resolution from 80 meters for Landsat 1 to 15 meters for the Operational Earth Resources Satellite (OERS) planned for launch in 1990 (Ref 1).^{*} In addition to the increased resolution, the OERS design currently calls for 20 different bands. This translates to an increase in total data rates for earth sensing missions from 1 million bits per second (Mbps) to over 100 Mbps, an increase of two orders of magnitude in less than five years. In addition to affecting the data rates, increased sensor resolution affects the following areas:

- o Greater processing requirements for image correction, formatting, and information extraction.
- o Increased navigation accuracy requirements to provide the necessary image distortion coefficients.
- o Increased archiving requirements to accept the larger data volume.

Increased resolution of the science sensor is a positive trend for applications. However, the acquisition of these data is totally non-deterministic in that nothing is known about the quality, content, or location of the imagery prior to or even shortly after it is obtained. As a result very little of the information is used by the scientific community because of such undesirable effects as cloud coverage, unwanted scene content, or exposure dates that do not coincide with those desired. In fact, less than 1% of all previously acquired remotely sensed data have been examined by the user.

In addition to the data deluge problem, it is currently not feasible to exploit Landsat data for such real-time applications as forest fire detection and monitoring and flood detection. In fact, with the exception of applications in which the observables do not change dramatically with time (such as oil exploration), Landsat's usefulness is limited from an operational standpoint because the data are already stale by the time the user gets them.

User requirements are another important trend associated with remote sensing missions because in the long run these requirements play a major role in defining the mission. Current data dissemination techniques are limited to a several-month turnaround. Many users require this time to be cut to hours, and a set of future users are requesting real-time control of the science instrument. The primary limiting factor in data turnaround time is the processing required for image registration. It is important to emphasize that not all users require real-time instrument control but the trend indicates a strong requirement for faster turnaround. For applications in which the primary observable does not change dramatically with time, or when the extraction of meaningful information is not time-critical, automated archiving techniques may be used in place of real-time instrument control.

^{*}References are given in Chapter VII.

The final trend that must be overcome to realize future missions is the rapidly escalating cost of ground support. The primary drivers of this trend are manpower costs and sophisticated equipment. A detailed examination of the problem reveals that the following areas are the most critical limitations of current remote sensing missions.

- o Remote sensing missions do not employ any automated techniques of data evaluation for either simple annotation or data negation. A simple determination of percentage cloud coverage in a scene could effectively eliminate on the order of 50% of the data acquired.
- o Users are subject to the schedule of the mission rather than the mission being tailored to the specific acquisition requirements of the user. This results in the acquisition of tremendous volumes of unwanted imagery either due to poor quality caused by atmospheric effects, the geographic location, or the time of acquisition.
- o All data, including tracking data from remote stations, must be sent to a central facility for image correction, annotation, and packetization, and then sent to a second facility for archiving and eventual dissemination to the users. This results in an unnecessary data link and awkward procedures to ensure critical timing requirements are met. This process is neither cost nor time effective.
- o No advantage is being taken of the onboard navigation capability to limit the magnitude of image distortions. By providing onboard control, the magnitude of the image distortions and hence the magnitude of the processing involved can be dramatically reduced.

Automation promises to be a key driver in overcoming the many limitations associated with current remote sensing missions. Areas that could benefit from this technology include:

- 1) Spacecraft control;
- 2) Spacecraft command generation and verification;
- 3) Data unpacking;
- 4) Image correction;
- 5) Data archiving;
- 6) Data dissemination.

The most critical issue associated with automation technology seems to be whether the data processing should be performed aboard the spacecraft or by ground support. The answer to this issue is complex and will involve many tradeoffs, but in the end it boils down to a question of economics and the long-term benefits and limitations of each approach.

In summary, the two most critical limitations of existing remote sensing missions are the deterministic acquisition of data, and the rapid correction of the data for radiometric effects and image distortions. The purpose of this contract was to investigate a new approach to remote sensing that would meet future mission requirements by overcoming these problems.

The primary sources of image distortion are sensor-peculiar errors, viewing perspective, coordinate transformation errors, and spacecraft-induced errors. With the development of the multilinear array, the primary sensor-caused distortions will be the individual placement of the detector elements, optical effects, and orientation of the array relative to the sensor coordinate frame prior to flight. These errors remain fairly constant over time so the resulting distortions are deterministic.

In referring to Figure I-1, viewing perspective is a well-known function of local earth radius and vehicle altitude and can be computed as shown. It is assumed that the local earth radius is known so this distortion is also deterministic.

The primary error source remaining, therefore, is spacecraft-induced. The spacecraft error sources can be categorized as:

- 1) Attitude determination,
 - a) Star tracker accuracy,
 - b) Star tracker configuration,
 - c) Knowledge of star tracker misalignment,
 - d) Error in star catalog,
 - e) Gyro noise,
 - f) Uncertainty in gyro bias, nonorthogonality and misalignment,
 - g) Numerical accuracy;
- 2) Ephemeris prediction,
 - a) Global positioning system (GPS) accuracy,
 - b) Dynamic model accuracy,
 - c) GPS update interval,
 - d) Numerical accuracy;
- 3) Transformation error between inertial and earth-fixed coordinates,
 - a) Knowledge of UT1,
 - b) Knowledge of earth precession, nutation, polar wander, and tidal deformation (note these error sources are generally included in GPS accuracy),
 - c) Numerical accuracy;
- 4) Misalignment between sensor and body coordinates,
 - a) Knowledge of linear array orientation,
 - b) Accuracy of thermal distortion model,
 - c) Vibration modes between two coordinates,
 - d) Calibrator technique and frequency,
 - e) Numerical accuracy.



$$\theta = 180 - \delta$$

$$D_i = \text{surface distance of } i^{\text{th}} \text{ element from the center of the field of view}$$

$$= \psi R_{\epsilon_i}$$

I-4

For the sake of discussion, assume that all the error is due simply to the attitude determination system. To achieve the temporal registration requirement of 15 meters, it will be necessary to predict attitude to within 4 arc-seconds as illustrated in Figure I-2. The accuracy of the state-of-the-art systems using the NASA standard star tracker and gyro is 15 arc-seconds (2σ) (Ref 2).

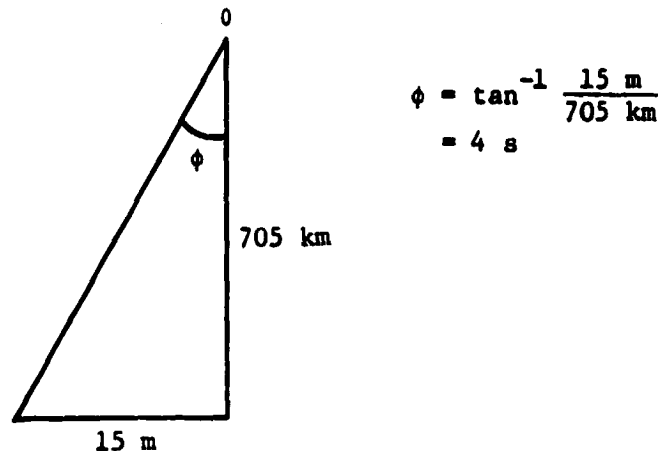


Figure I-2

Error Budget for Registration Accuracy of 15 m

Even with the development of charge-coupled device (CCD) star trackers, the attitude determination capability will be around 6 arc-seconds (2σ). Note that 2σ numbers have been used here corresponding to 95% of the data. If 1σ numbers corresponding to 67% of the data are used, the accuracy goals can be met. However, by adding even one more error term such as misalignment between sensor and body coordinates of 2 arc-seconds (2-axis alignment accuracy achievable with optical alignment cubes), the total error budget is exceeded. From the previous discussion, which ignored many error sources, it is clear that another approach is required.

The key to real-time image correction lies in being able to accurately determine the location of the science sensor's boresight in earth-fixed coordinates. With this capability it is not only possible to provide real-time measurements of the image distortions, but with the advent of the multilinear array (MLA) it may be possible to provide real-time image correction using either resampling or special design of the MLA focal plane. This capability also provides the heart of a pointing system capable of deterministically acquiring imagery at specific earth-fixed coordinates.

Shortly after definition of the feature identification and location experiment (FILE), Martin Marietta began the development of a landmark tracker or GCP detector centered around experience gained with terminal guidance systems. The primary purpose of the landmark tracker is to provide periodic measurements of the science sensor's boresight position to be used as input to a navigation system. Previous studies (Ref 4 thru 7) have shown that the landmark tracker cannot solve for both

position and attitude without supplemental measurements from another source. Another limitation of the landmark tracker operating in the visible spectrum is that observations can be obscured by clouds and the correlator will lock onto a false target. For these reasons, the conceptual navigation system consists of the landmark tracker, a GPS receiver, and two NASA standard star trackers to replace attitude measurements when the landmark is obscured.

The purpose of this contract was to simulate operation of a navigation system centered around image correction, and analyze performance of the system under a variety of conditions. Also of interest was a sensitivity analysis to determine the optimal sampling interval for each sensor, and the total number of landmarks required to satisfy mission requirements.

In the past there has been a lack of coordination between the scientific user community and the engineers responsible for spacecraft design. This has resulted in a physical separation between the design and implementation of the science payload and the control system. An example of this thinking is shown in the multimission spacecraft (MMS) where subsystems are treated as modules and the payload itself is physically separated from the control portion of the vehicle. This type of design, although attractive from a standardization point of view, ignores the inherent relationships between user requirements, spacecraft control requirements, and ground support requirements. It is possible that rather than saving significant costs, standardization may result in higher end-to-end costs. This design philosophy must be reevaluated with regard to future missions. In the end-to-end design of remote sensing spacecraft, the primary emphasis in the guidance and control system must shift from simply estimating the ephemeris and attitude of the spacecraft to estimating the position of the science sensor's FOV on the earth's surface. This provides the basis for both real-time image correction and deterministic data acquisition through sensor pointing. This shift of emphasis will impact the design of the entire spacecraft. For example, if the science sensor boresight position is to be determined, it is desirable to place the gyro package close to that sensor to reduce the misalignment between the two. The (MMS) configuration, which provides a physical separation between the payload and the guidance and control system, may not satisfy the requirements of many future remote sensing missions.

This report has been separated into four volumes. The first volume entitled "Executive Summary" is intended to provide an overview of the image correction system, a brief examination of the analysis, and a discussion of the impacts on future missions. This volume, entitled "Analysis and Simulation Results," provides a detailed account of all work performed under the contract. Volumes III and IV contain a detailed description of all software developed and used in this contract.

This volume has been separated into the following chapters:

- I. Introduction;
- II. Onboard Image Correction System Overview;
- III. GCP Navigation System Simulation;
- IV. GCP Analysis;
- V. Image Correction Accuracy Measurement;
- VI. Conclusions and Recommendations;
- VII. References.

II. ONBOARD IMAGE CORRECTION SYSTEM OVERVIEW

The onboard image correction system has been separated into three primary subsystems as illustrated in Figure II-1. The navigation processor accepts measurements from the GPS receiver, the star trackers, and the landmark tracker, and solves for spacecraft state. The registration processor is responsible for correlating live imagery with stored reference ground control points to produce a landmark sighting vector. This processor is also responsible for providing measurements of the distortion coefficients, and for providing image resampling if this operation is performed on board. The pointing mount controller has different uses depending on the application. If real-time image correction is being provided through the use of a curved focal plane multilinear array (MLA), the pointing mount would be used to ensure that the instrument boresight always pointed at nadir. If selective acquisition is being provided, the pointing mount could be used to provide additional flexibility. This chapter describes each of these subsystems in greater detail.

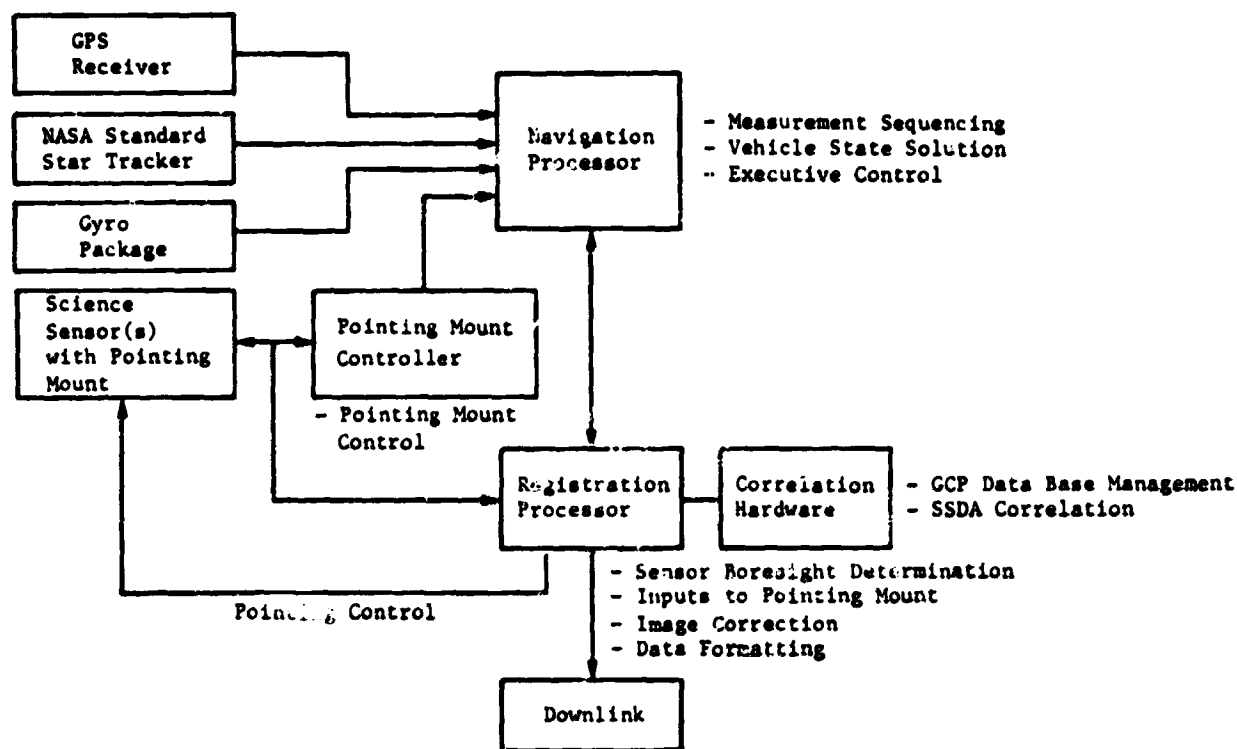


Figure II-1 Onboard Image Correction System Block Diagram

A. REGISTRATION PROCESSOR

Figure II-2 illustrates the functional flow of the registration processor. A digital representation of all the ground control points is stored on a high-density tape recorder (or bubble memory) and read into random access memory as required. Because of the reliability considerations associated with the recorder, a number of GCPs will be read into RAM at one time. As the spacecraft proceeds in orbit, the landmark that is to be viewed next is determined and the digital reference image is extracted from RAM. Information stored in the GCP file, along with uncertainty in vehicle position and attitude, helps to define the location and size of the search area. For an operational system it is advisable to use a fixed-size search area. Since a star tracker is being used for backup, the maximum attitude error will be in the vicinity of 15 arc-seconds (2σ) and the global positioning system will produce an error around 12 meters. Therefore a search area that is 32 pixels larger than the GCP size will provide far more area than required. For this reason, and the fact that a GCP size of 32 produces the best results, a search area size of 64 was chosen.

Proceeding with registration processing, a test is initiated to determine if the location of the present scan line is coincident with the position of the search area. If it is not, a new satellite state is obtained from the navigation processor. This new state is then transformed into sensor pointing information for use by the pointing mount controller for several purposes. First, if a curved focal plane array has been used to provide real-time image correction, the pointing mount must isolate the array from the spacecraft motion to ensure that the central element of the array is viewing directly along nadir. If the pointing mount is being used to provide deterministic data acquisition according to earth-fixed coordinates, the boresight position provided by the registration processor will be used as a reference to point from. Following the determination of boresight position, the loop continues until the current scan line is coincident with the search area.

At this point, image scan lines extracted from the sensor are stored in a buffer and become the search area data. The search area is then registered with respect to the GCP and the registration vector passed to the navigation processor, which uses this information to update the estimate of vehicle state. The entire process is then repeated for the next landmark. The following paragraphs describe in more detail the operations performed during each of these functions.

1. Organization of the GCP Data Base

The architecture of a GCP data base is closely related to the requirements placed on a particular mission. For the advanced Landsat-type mission model being considered, we selected a data base that is primarily a sequential access file but allows implementation of random access for missions requiring deterministic data acquisition through sensor pointing. The data base has been broken into four levels of directory as shown in Figure II-3.

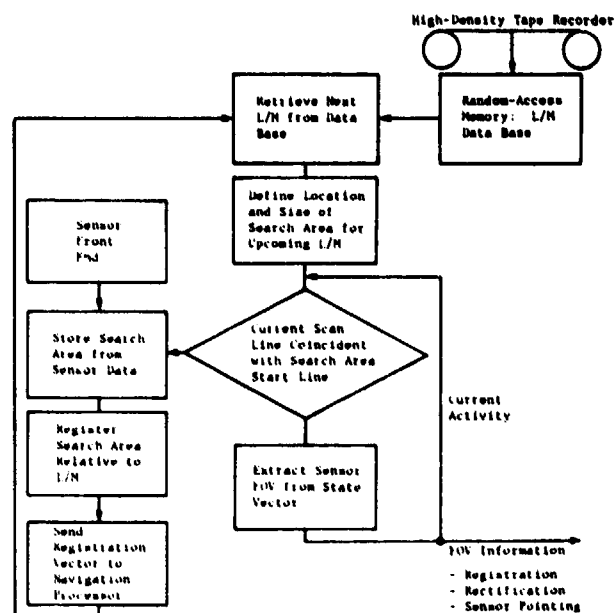


Figure II-2 Functional Flow of Registration Processor

The first level contains one entry for each orbit of the satellite. For Landsat, this would mean 16 days worth of entries since the orbit is repetitive with this period. Each entry in the orbit table can be broken down into sections of an orbit where the size of the section will be determined by equating the number of GCPs contained in a section with block size to be accessed from the recorder. Finally, an entry in the orbit section table can be broken down into the GCPs contained in that section. This final level of data base is called the count table and contains all the necessary information to locate the position of the search area plus the starting address in RAM containing the digital representation of the GCP.

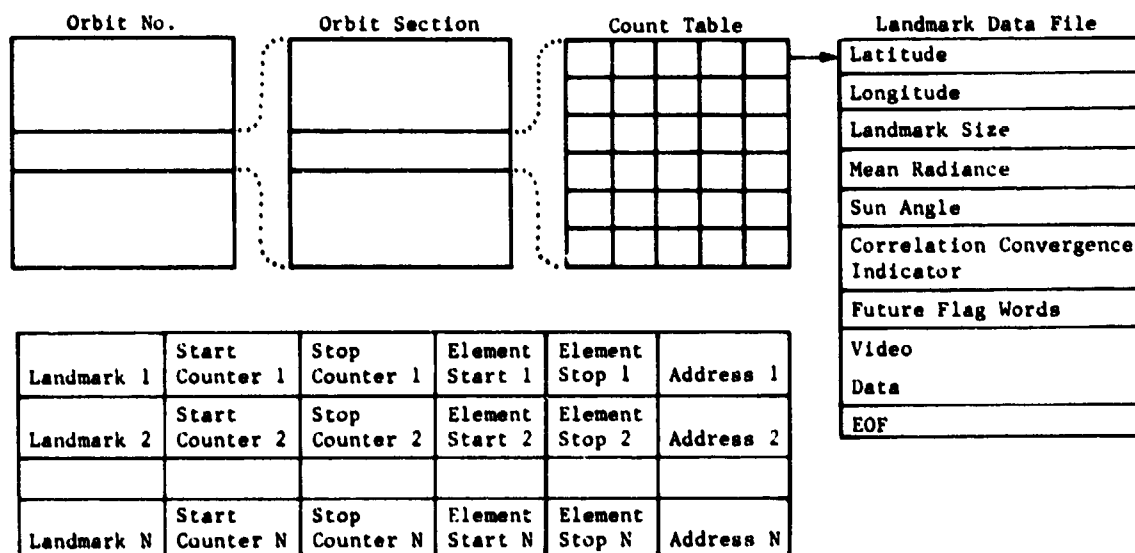


Figure II-3 Organization of the GCP Data Base

2. Determination of Sensor Boresight

The position of the science instrument boresight in earth-fixed coordinates is required for pointing control (Fig. II-4). In referring to Figure II-4, the boresight position may be found by forming a unit vector along the boresight in the sensor coordinate system

$$\underline{U}_L = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

where \underline{U}_L is a unit vector in landmark tracker coordinates. This vector may then be transformed into inertial coordinates through

$$\underline{U}_I = I_{TB} B_{TL} \underline{U}_L$$

where

\underline{U}_I = unit vector along the sensor boresight in inertial coordinates,

B_{TL} = transformation from sensor coordinates to body coordinates accounting for pointing angle offsets,

I_{TB} = the direction cosine matrix transformation describing the attitude of the vehicle.

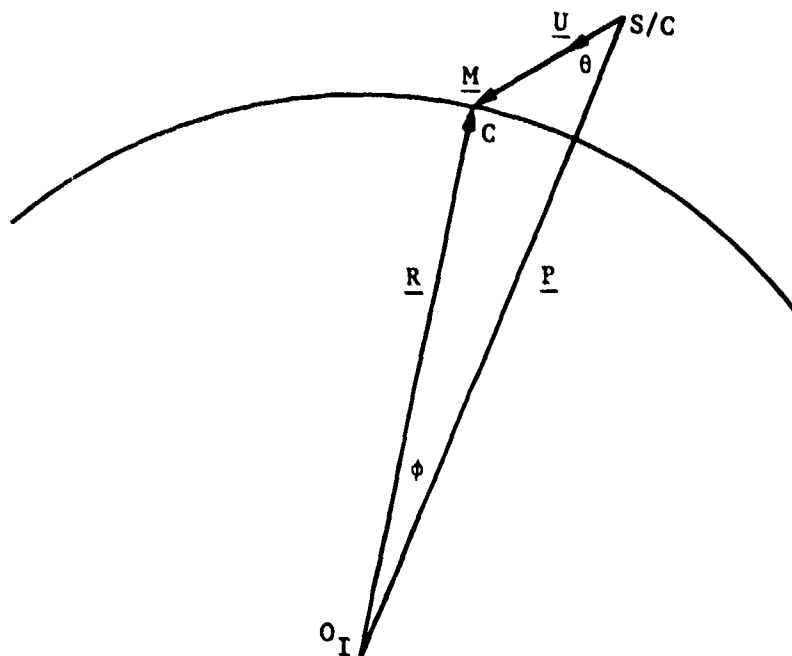


Figure II-4
Determination of Sensor Boresight Surface Position

Since

$$\underline{R} = \underline{P} - \underline{M}$$

Where

\underline{R} = radius vector from the origin of the earth-centered inertial coordinate system to the boresight surface position,

\underline{P} = space craft position vector in ECI coordinates,

\underline{M} = measurement vector from the spacecraft to the boresight position on the earth's surface.

It is necessary therefore to define \underline{M} . This may be done using the laws of sines and cosines as follows. With $|\underline{R}|$, the magnitude of \underline{R} known, ϕ may be determined by the following process.

$$\frac{|\underline{R}|}{\sin \theta} = \frac{|\underline{P}|}{\sin C} = \frac{|\underline{P}|}{\sin (180-C)}$$

Where θ is determined by

$$\theta = \cos^{-1} \frac{\underline{U} \cdot \underline{P}}{|\underline{U}| |\underline{P}|}$$

Solving for C in terms of θ , \underline{R} and \underline{P} produces

$$C = 180^\circ - \sin^{-1} \frac{|\underline{P}| \sin \theta}{|\underline{R}|}$$

Since

$$\phi + C + \theta = 180^\circ,$$

$$\phi = 180^\circ - C - \theta = \sin^{-1} \frac{|\underline{P}| \sin \theta}{|\underline{R}|} - \theta.$$

Using the law of cosines

$$\begin{aligned} |\underline{M}| &= (|\underline{M}|^2)^{\frac{1}{2}} = (|\underline{R}|^2 + |\underline{P}|^2 - 2|\underline{P}||\underline{R}| \cos \phi)^{\frac{1}{2}} \\ &= (|\underline{R}|^2 + |\underline{P}|^2 - 2|\underline{P}||\underline{R}| \cos (\sin^{-1} \frac{|\underline{P}| \sin \theta}{|\underline{R}|} - \theta))^{\frac{1}{2}}. \end{aligned}$$

Since \underline{M} is directed along \underline{U} ,

$$\underline{M} = |\underline{M}| \underline{U}$$

and

$$\underline{R} = \underline{P} - \underline{M}.$$

The longitude (L) and latitude (λ) of the surface position may be determined if \underline{R} is transformed into earth-fixed coordinates. This transformation, $E T_1$, is a function of time, earth rotation, polar wander, nutation, etc and is assumed to be known to the system.

Therefore

$$\underline{R}_E = E^T I \underline{R}_I = \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix}_E$$

Where

R_i = the projection of R_e along the i axis, for $i = x, y, z$ and

The respective earth-fixed coordinate axes consist of z along the earth's nominal spin axis, x from the earth's center through the intersection of the equator and the Greenwich meridian, and Y completes the right-hand set.

The longitude and latitude of the boresight surface projection are given by

$$\lambda = \sin^{-1} \frac{R_z}{|\underline{R}|}$$

$$L = \sin^{-1} \frac{R_y}{|\underline{R}| \cos \lambda}$$

3. Extraction of Search Area Data

The location and size of the search area is completely defined by the line start, line stop, element start, and element stop indicators contained in the count table as shown in Figure II-5.

Present Sensor
Scan Line

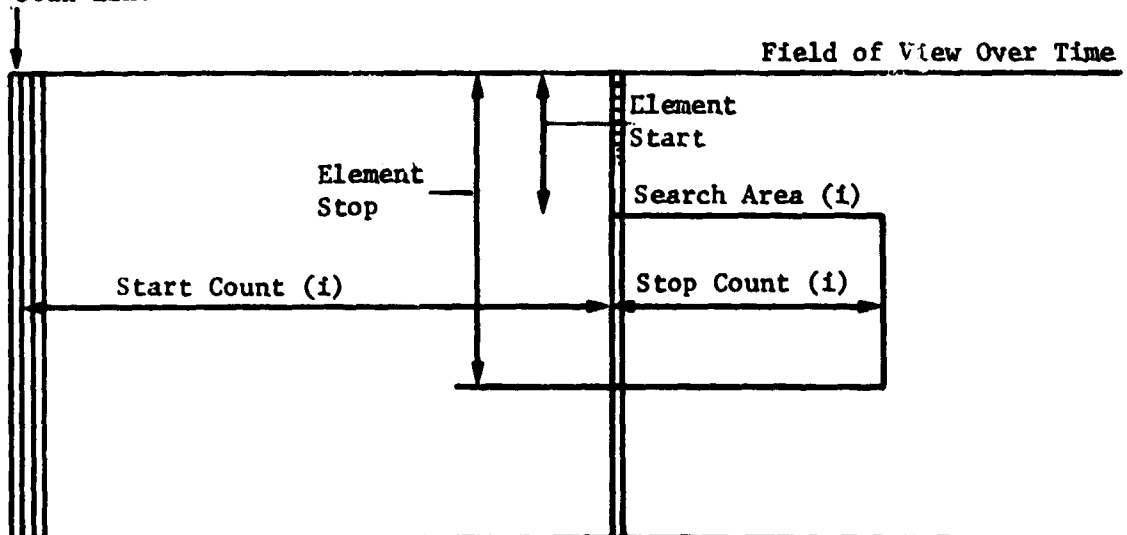


Figure II-5 Explanation of Parameters in Count Table

The two long horizontal lines enclose a diagram of the imaging sensor's field of view over time. The vertical swaths depict one scan of the sensor's instantaneous field of view, and the small squares within the scan line are individual picture elements. The start count contains the number of scans between the present scan line and the first line of the search area. The stop count contains the size of the search area in scan lines. Once it is known that the search area is within the field of view, it is necessary to determine its location. The element start count contains the distance, in picture elements, of the search area from the start of the scan line. The element stop count contains the distance to the end of the search area from the start of the scan line. These four parameters totally describe the position and size of the search area relative to the current position of the sensor's field of view. For a constant search area size, two of the parameters may be eliminated to reduce the onboard storage requirements. The center of the search area is coincident with the estimated center of the landmark and its size is a direct function of the uncertainty in satellite position and attitude. To accommodate the varying size of the search area over time, the parameters in the count table can be changed.

The algorithm implemented to extract the search area from the sensor's field of view (FOV) is shown in Figure II-6. The start count indicator is decremented each time a scan line is read by the sensor. The decrement process is driven by the same clock signal that drives the scan mechanism so the counter does not diverge. When the counter reaches zero, the search area has been encountered and the second loop in the figure is entered. In this loop image data from the sensor are read into a buffer each time a scan is initiated. The buffer does not contain the entire scan line, but only the data between the element start and element stop flags previously described. After a scan line has been recorded, the element stop count is decremented. When this counter is zero, the entire search area has been read and the correlation process can begin.

4. Registration of Sensor Data

The registration processor is centered around a sequential similarity detection algorithm (SSDA) first identified by Barnea and Silverman (Ref 8). Various correlation algorithms were analyzed under a separate contract (Ref 9) and the results of this study indicate that the SSDA can be used to detect features to within one-tenth the resolution of the science instrument. In addition the SSDA was found to be much simpler to implement in an operational system primarily because it consists of simple additions whereas other algorithms involve multiplications and powers of large numbers.

A ground control point may be defined as a small area within the sensor's FOV whose characteristics are easily detected using a registration algorithm such as the SSDA. It is not necessary to process imagery from the entire FOV, but only an area whose size ensures that the GCP will be located within its boundaries. The size of this search area is a direct function of the uncertainty in the vehicle's position and attitude. However, for an operational system where timing requirements are critical, it is advisable to determine the maximum expected error and to fix the search area size.

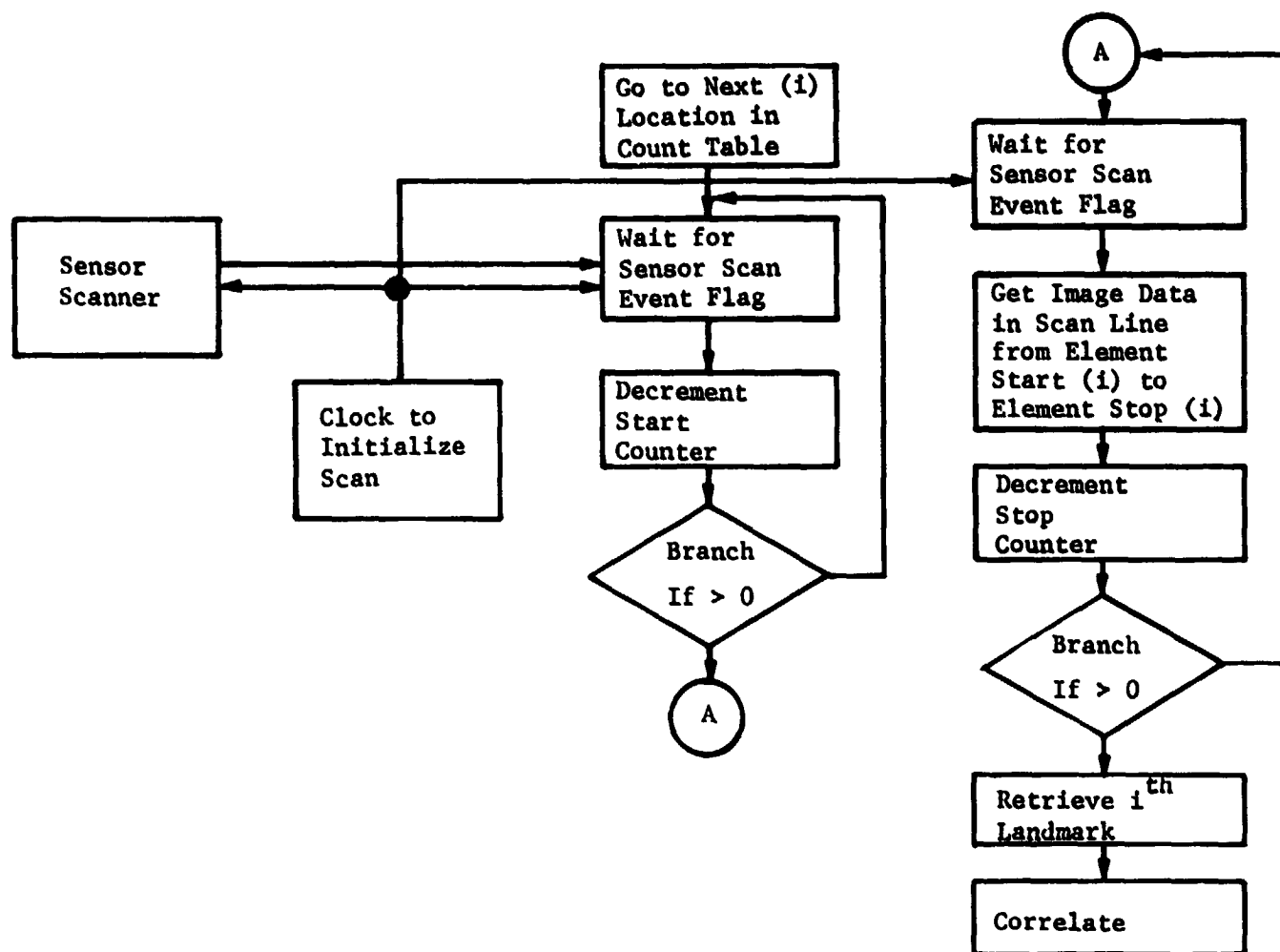


Figure II-6 Search Area Algorithm

The registration process may be described in the following way. Let the search area be defined as an $L \times L$ area of digital picture elements. The imagery may then be defined by a function, S , that describes the gray scale, or recorded radiance, in relation to position coordinates, i.e.,

$$S(i,j) = W_{i,j}$$

where $W_{i,j}$ is the gray scale of the i,j^{th} picture element of the search area defined by

$$1 \leq (i,j) \leq L.$$

Let the ground control point be defined similarly as an $M \times M$ area with an image function

$$G(l,m) = R_{l,m}$$

where $R_{l,m}$ is the gray scale of the l,m^{th} picture element of the GCP defined by

$$1 \leq (l,m) \leq M.$$

A subimage (Fig. II-7) of the SA may be defined as an $M \times M$ area whose upper left coordinates (n,o) lie in the range

$$1 \leq (n,o) \leq L - M + 1.$$

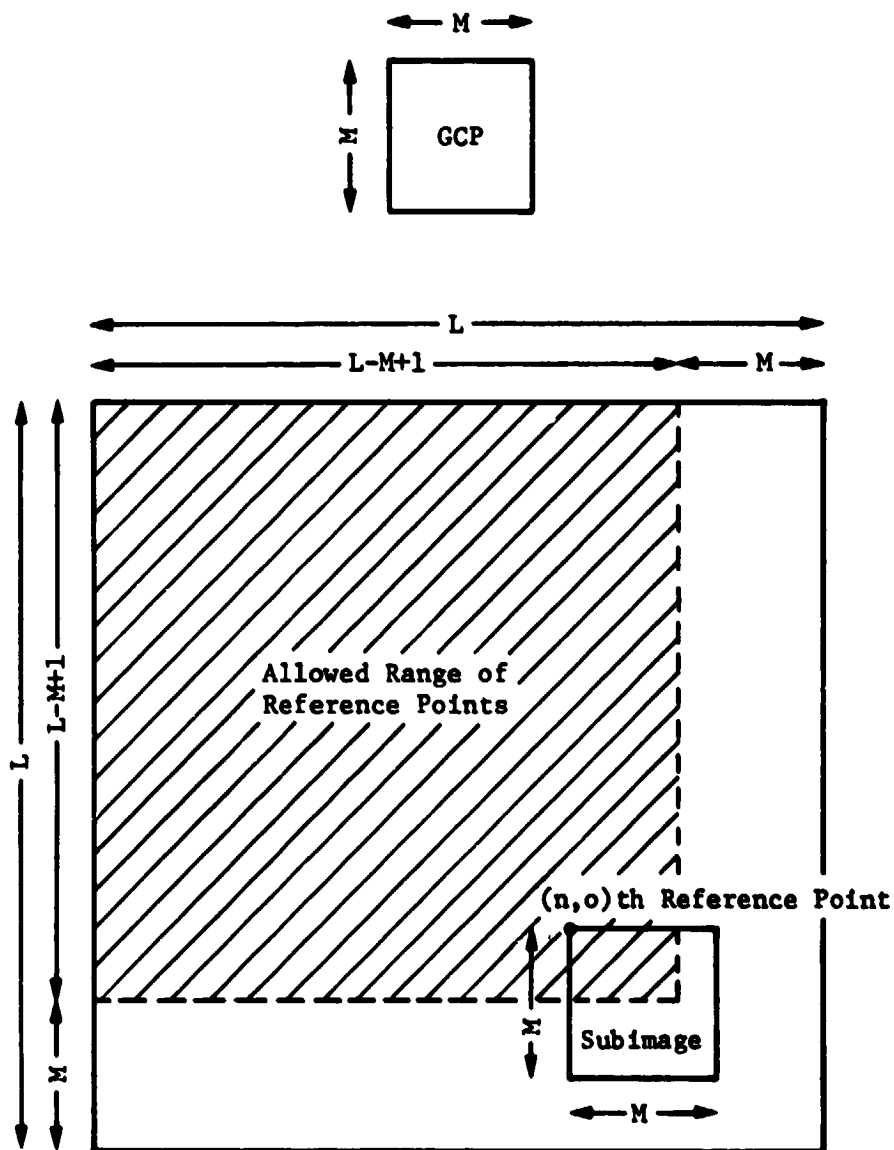


Figure II-7 Subimage Definition

A subimage whose upper left coordinates are n,o will be referred to as the n,o^{th} reference point.

The sensor data is registered by measuring the similarity between each MXM subimage within the search area and a digital representation of the GCP stored onboard. The reference point that produces the highest degree of similarity with the GCP is the best registration of the GCP and can be labeled with the same earth fixed coordinates as the GCP.

The SSDA algorithm may be implemented to detect similarity between a reference point and the GCP through the following equation:

$$\text{Similarity} = \sum_{i=1}^M \sum_{j=1}^M \left| \left[S(i+n, j+o) - \bar{S}_{n,o} \right] - \left[G(i,j) - \bar{G} \right] \right|$$

Where

S_{no} = n,oth Reference Point,

\bar{S}_{no} = Mean value of the subimage located at the nth reference point,

\bar{G} = Mean value of the GCP

The entire registration process can then be described by the algorithm shown in Figure II-8.

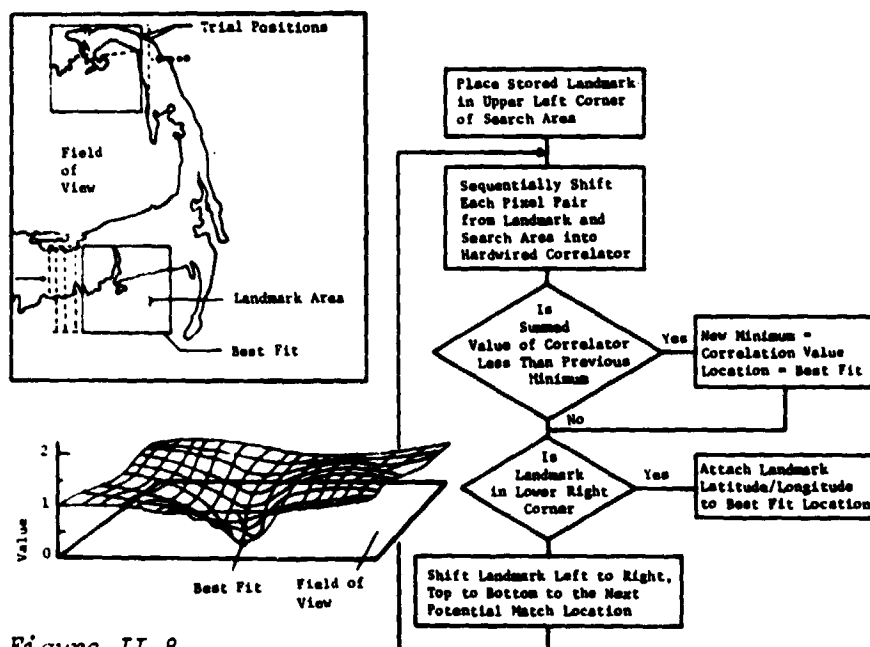


Figure II-8

Registration of Sensor Data via GCP Detection

Some evidence suggests the mean value of separate features should be independently computed. For example, the mean radiance of vegetation would be computed separately from bare earth, and this mean would only be used to normalize vegetation pixels within the subimage.

Several approaches to subpixel registration were evaluated under independent research. The first approach consisted of interpolating the imagery corresponding to the reference point that produced the highest degree of similarity. The interpolation of the imagery consisted of resampling the imagery at a 10:1 scale. The resampled image was then registered using the SSDA algorithm with the GCP. This technique

proved to be useful in providing a registration accuracy to within 0.1 Pixel (Ref 10). However, the processing involved in resampling the image was found to be excessive for an onboard system. Therefore a second approach was considered.

As similarity is being measured by the SSDA the correlation surface is stored in a separate array. When the best fit is located, the surface is interpolated by fitting a cubic spline, and the minimum value of the function is obtained. Results obtained show that this technique is also capable of providing a registration vector with an accuracy of 0.1 Pixel. The processing involved in interpolating the surface is also well within the capabilities of an onboard system.

5. Cloud Suppression

A small percentage (10%) of cloud coverage within the search area can affect the correlation process in such a way that the GCP is no longer recognizable. This effect can be minimized by simply eliminating every picture element representing a cloud from the correlation and normalizing the coefficient according to the number of pixels correlated. Preliminary results indicate that 20 to 40% cloud coverage can be tolerated depending on the type of landmark and what portion of the scene is obscured. The problem then becomes one of detecting clouds in the scene onboard in real time.

The feature identification and location experiment (FILE) system provides the capability of classifying the sensor field of view into the four generic features of bare land, vegetation, water and clouds. Figure II-9 illustrates how various feature types can be separated using two sensors operating at 0.65 and 0.85 μm .

Although the polygons in the figure overlap to a certain extent because they represent 99% confidence limits, the majority of cases can be expected to fall in unambiguous points on the plane. This technique was incorporated into the concept for onboard GCP detection to provide a pixel-by-pixel suppression of clouds.

6. False Lock Detection

Since the registration vector is being used to update the vehicle navigation filter, it is of the utmost importance to detect when the correlator has locked onto a false target. Analysis indicated that correlation surfaces resulting from false target lock always converged more slowly than those resulting from a correct lock. The first approach to false lock detection was to measure the convergence rate of the surface and implement a threshold discrimination algorithm. However, the convergence rate was found to be a complex function of several variables that included target contrast and geometry, indicating that a simple threshold algorithm would not perform well for all cases. A more thorough analysis showed that a better indication of false lock would be to measure the similarity between the correlation surface obtained from registration of the search area with respect to the GCP and the correlation surface obtained with GCP autocorrelation. The false lock coefficient is measured using a classical correlation algorithm.

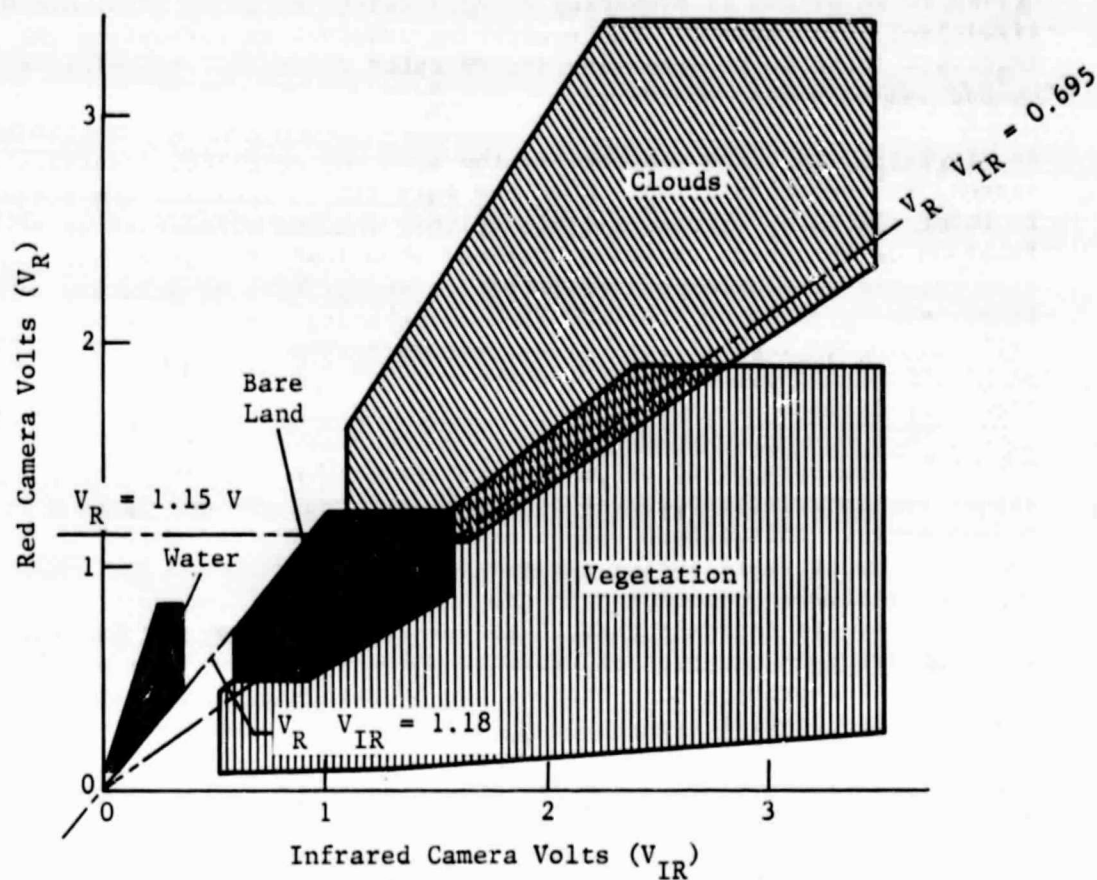


Figure II-9 99% Confidence Polygons for Feature Classification

The accuracy of false lock detection was studied under the analysis and results indicate that 99% of false locks may be avoided.

7. Correlator Design

In an operational system, it is highly desirable to implement much of the registration processor functions in a hardwired system. This section discusses one possible implementation of the correlator.

There are four elements in the landmark correlator:

- 1) Data selection;
- 2) Data buffering;
- 3) Correlator;
- 4) System control.

The following paragraphs describe each of these elements.

a. Data Selection - Data selection is a two-step pipeline operation. Data are assumed to be generated by the sensor as an 8-bit wide parallel data stream. Each 8 bits define the value of one pixel. Two synchronization signals are also assumed to be available from the sensor. These are DATA CLOCK and beginning of line (BOL). DATA CLOCK is a pulse that indicates that the data lines contain data. BOL indicates the time at which the sensor is starting a sweep or scan.

Figure II-10 illustrates the parameters involved in the data selection process. The area in which the landmark resides is smaller than the sensor field of view. The landmark area is defined LINE COUNT by PIXEL COUNT, or number of lines by number of pixels. Each line is assumed to contain the same number of pixels. The symbolic upper left-hand corner of the landmark area is defined by the values LINE BIAS by PIXEL BIAS. Some synchronization point must be supplied for the LINE BIAS measurement. This synchronization point could be as simple as a timer supplied by the landmark system control element. The synchronization point for PIXEL BIAS is the BOL signal generated by the sensor.

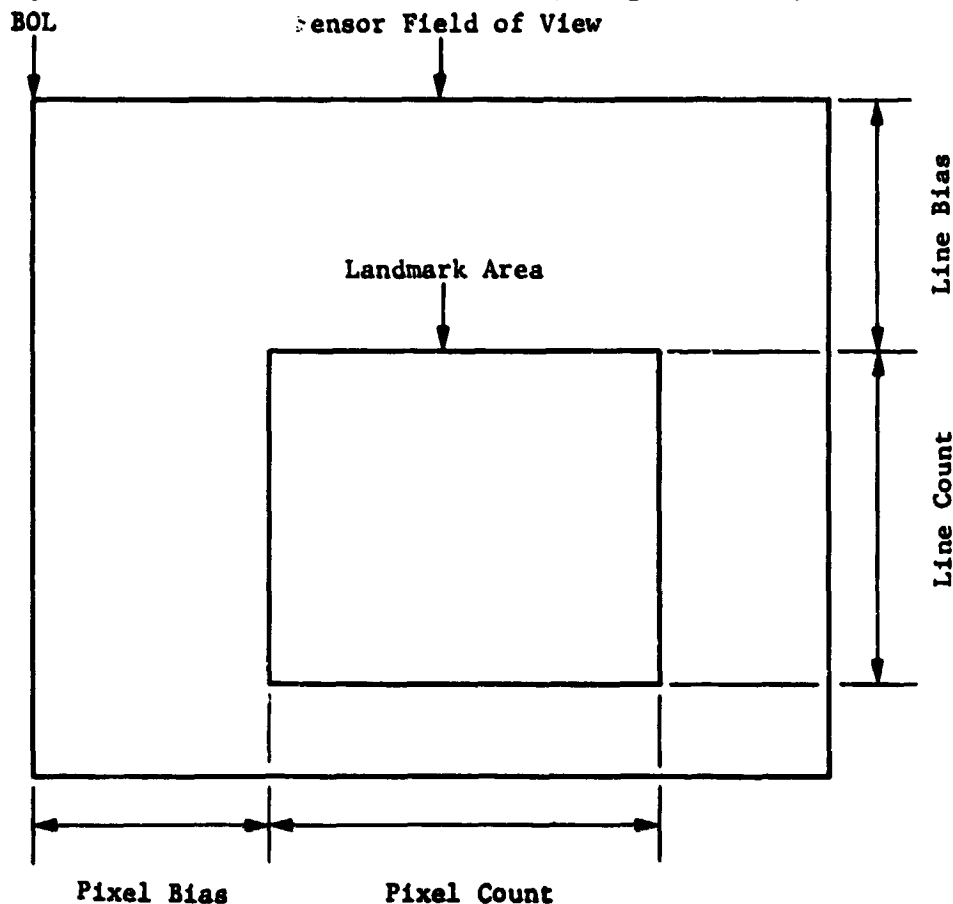


Figure II-10 Parameters Involved in Data Selection Process

The first step in the data selection process is to select the proper scan lines. This is accomplished by initializing two counters with the values of LINE BIAS and LINE COUNT. Each time a BOL signal is generated by the sensor, the LINE BIAS counter is decremented by one. When the LINE BIAS counter reaches zero, the signal LSTRT is generated, indicating the first line of the landmark area. At this time the counter

containing LINE COUNT will be decremented for each BOL pulse. When the LINE COUNT reaches zero, LSTOP is generated. A simple latch is utilized to control an LSAVE signal that denotes the time from LSTRT to LSTOP. This is the time when sensor scan lines correspond to the landmark area.

The LSAVE signal enables the second step of the data selection process. This step works in the same way as the first. Instead of counting lines, however, pixels from BOL are counted. The output of the select proper pixels step is the PSAVE signal indicating that pixels within the landmark area are being generated by the sensor. This PSAVE signal is used to drive the next landmark correlator element, the data buffering element.

b. Data Buffering - As would be expected, data buffering involves both an input stage and an output stage. The input stage is started by the PSAVE signal. When this signal is true, data from the sensor are shifted into a RAM buffer in a direct memory access (DMA) fashion. The DMA logic is implemented by two down-counters, one counter controlling the RAM transfer address and the other counter controlling the transfer count.

The output stage of data buffering may begin before the entire landmark area data have been collected because the correlator element will operate in a serial pipeline manner. Unfortunately the order in which data are input to the RAM buffer is not the same order in which data are output from the RAM buffer. For example if the landmark area is dimensioned as 40 lines by 40 pixels, the incoming data stream is

$L_1P_1, L_1P_2, L_1P_3, \dots, L_1P_{39}, L_1P_{40}, L_2P_1, L_2P_2, \dots, L_{40}P_{40}.$

If the landmark is 16 lines by 16 pixels, the data stream output from the RAM buffer will be

$L_1P_1, L_1P_2, \dots, L_1P_{15}, L_1P_{16}, L_2P_1, L_2P_2, \dots, L_{16}P_{16}$

to compute $W_m (1:16, 1:16)$ and $C(1:16, 1:16)$, and

$L_1P_2, L_1P_3, \dots, L_1P_{16}, L_1P_{17}, L_2P_2, \dots, L_{16}P_{17}$

to compute $W_m (1:16, 2:17)$ and $C (1:16, 2:17)$, etc. Each output data stream from the RAM buffer may be handled in a DMA mode but the system control element must continually reset the initialization logic of the output stage to ensure correct data selection.

c. Correlator - The correlator is implemented as a four-step pipeline. The required steps are:

Step 1: Compute $\bar{W}_m = \frac{\sum_{i=m}^n \sum_{j=m}^n W_{ij}}{n - m}$.

Step 2: Compute $C_x(i,j) = W_m + L_x(i,j)$
where $L_x(i,j) = L_m(i,j) - L_m$

Step 3: Compute $C_y(i,j) = W_m(k,l) - C_x(i,j)$,

Step 4: Compute $C = C + C_y(i,j)|$.

With this approach it is necessary that step 1 be completely finished before the other steps can begin. Once this is done, however, steps 2, 3, and 4 may operate simultaneously with step 3 one data point behind step 2 and step 4 one data point behind step 3. The output of step 4 is the correlation value for one position of the landmark within the landmark area. Computation of all the correlation values is tied in with the output stage of the data buffering element and this is described more fully in the following paragraph.

d. System Control - Because of performance requirements the three elements just described (data selection, data buffering and the correlator) should be implemented using hardwired logic. Hardwired logic could also be used for the system control element although this appears neither necessary nor desirable at this time. A much more attractive approach could be the use of a dedicated microcomputer. The system control functions can be described quite easily in software and indications are that obtaining sufficient performance would not be a problem. The system control functions required are:

- 1) Data buffering input stage initialization;
- 2) Data buffering output stage initialization;
- 3) Correlator step 1 and 2 synchronization;
- 4) Correlator step 4 data acceptance.

As data is collected from step 4 of the correlator the system control element must keep track of the minimum C value and where this value occurs. This information is then furnished to other processes for determining spacecraft state.

e. Landmark Correlator, Performance Estimate - Figure II-11 shows a top-level block diagram of the elements just discussed. Several approaches were considered and some were immediately discarded because sufficient performance could not be obtained. Other approaches involving more parallel operations could be used to obtain even higher performance but the additional hardware complexity did not appear justified at this time.

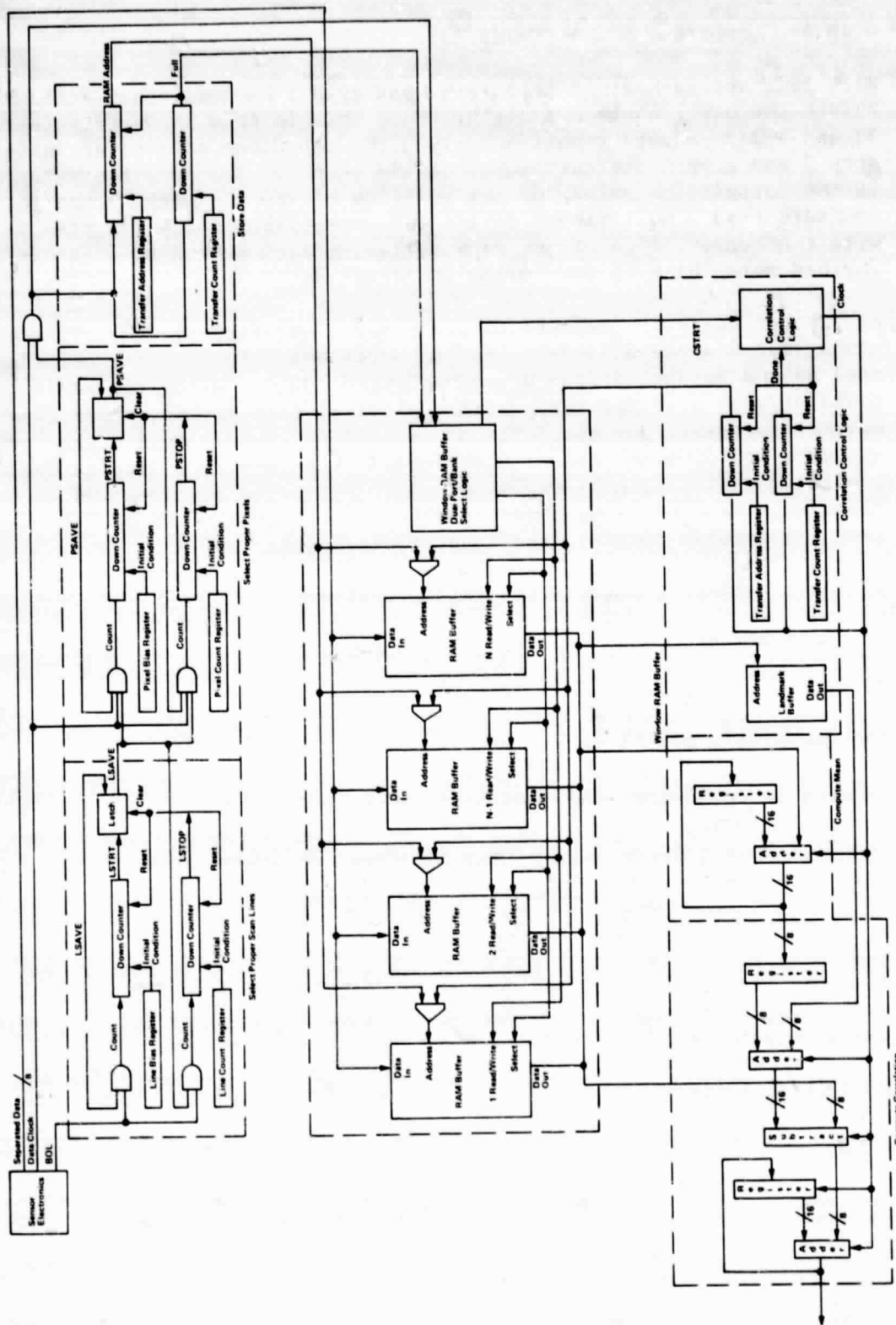


Figure II-11

Figure II-11 Correlator Design

Two types of technology were considered for hardware implementation--TTL and CMOS/SOS. In both cases the limiting performance element was step 3 of the correlator. For the TTL implementation a maximum clock* of 10.5 MHz could be obtained. If semicustom CMOS/SOS LSI is used, this same element has a maximum clock* frequency of 12 MHz and has the additional advantage that it requires significantly less power and component count.

With the step 3 correlator limited to 12 MHz, one correlation coefficient can be computed every 22 μ s for a 16x16 landmark. If the landmark area is 40x40, correlation of the entire area requires 11 μ s. Since the original goal was to do the whole correlation in less than 1 second, this performance level was considered adequate.

B. NAVIGATION PROCESSOR

Figure II-12 illustrates the functional processing performed by the navigation processor. The most critical function performed is the sequencing of measurements and supervisory control of the registration processor. The sequencer first decides, based on a measurement profile, what type of measurement should be made and when the measurement should be made. The types of measurements the sequence tables establish are star trackers 1 and 2 and GPS. After determining the time of the next measurement, the sequencer decides, based on the total number of scan lines left to the landmark and the scan rate of the sensor, whether there is enough time before the landmark sighting. If there is, the measurement is processed normally. If there is not, control is defaulted to the registration processor until the registration vector is returned.

After the measurement is obtained, the estimated state vector is integrated forward to the measurement time. This process is part of the extended square root navigation filter but is implemented in a separate package to simplify control. The raw measurements are processed to compensate for knowledge of such error terms as bias and misalignment. These compensated measurements are used as input to the Kalman filter package that then estimates the spacecraft state. The spacecraft state includes:

- 1) Position;
- 2) Velocity;
- 3) Attitude;
- 4) Gyro Drift, nonorthogonality, and scale factor;
- 5) Science sensor misalignment.

*Maximum clock frequency derated 50% for spaceborne environment.

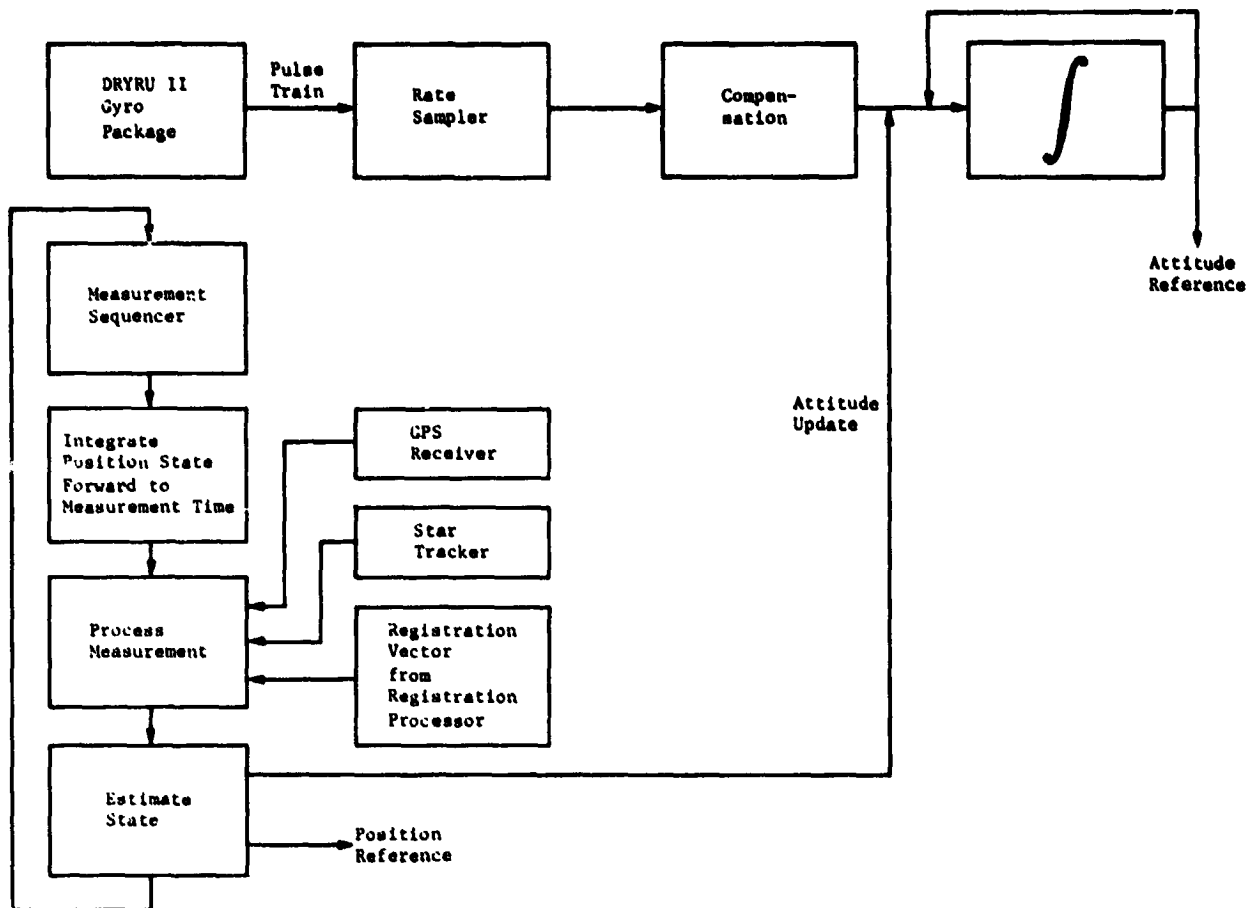


Figure II-12 Navigation Processor Functional Description

The navigation processor is responsible for sending the navigation state to the registration processor at the beginning of each scan line. This information is used to compute the distortion coefficients and to control the sensor pointing mount. Attitude is propagated between measurements by a NASA standard gyro package. The pulse train output from the gyros is compensated for knowledge of bias, nonorthogonality, and scale factor and then integrated to provide vehicle attitude.

Each of these processes is described in more detail in Chapter IV under the sections discussing dynamics and integrator, sequencer, Kalman filter, and gyro processing.

C. POINTING MOUNT CONTROLLER AND LINEAR ARRAY

The pointing mount controller can be implemented in two different ways, depending on the configuration of the image correction system. If a curved focal plane array is used, the pointing mount controller would be responsible for maintaining the boresight of the instrument along

nadir. In a system where image correction is provided through resampling, the pointing mount controller could be used to provide deterministic data acquisition by pointing to specified earth-fixed coordinates.

In a curved focal plane configuration, the pointing mount controller would form an error vector that represents the angular offset between the boresight of the linear array and nadir. This error vector would then be used as an input to the pointing mount to correct for the error. In this way, the viewing angle of the MLA would be isolated from the spacecraft motion caused by either attitude limit cycles, or attitude corrections.

In a selective acquisition mode the sensor boresight position would be compared to the desired pointing angle to generate an offset that would be used as an input to the pointing mount. Again the viewing angle of the sensor can be isolated from the spacecraft motion to simplify the image correction process.

The idea behind a curved focal plane array originates from the understanding that, with the development of the MLA, the primary image distortions will be caused by the viewing angle and the curvature of the earth (Fig. II-1). This nonuniform sampling is relatively constant and is symmetrical about a vector pointing along nadir. It was therefore felt that constructing an array with nonuniform sampling for each element that compensated for these effects could eliminate along-scan resampling, which is one of the largest bottleneck of remote sensing (Fig. II-13).

The limitations of this approach are that the central element must always point at nadir to within 0.5 the resolution angle of the sensor. While this is anticipated as being feasible with either Gimballflex or the angular suspension pointing system, a constraint is placed on the system in that it can no longer be used for deterministic data acquisition through sensor pointing. On the other hand, the cost savings and increased data turnaround time make this an attractive possibility that warrants further investigation. The key considerations that need to be addressed are:

- 1) Can the curvature of the focal plane maintain a precision that ensures additional distortions are not introduced?
- 2) Is the variation in the local curvature of earth great enough to introduce errors greater than 0.5 pixel?
- 3) Are variations in the orbital altitude great enough to cause distortions?
- 4) What is the cost savings of performing real-time image correction using this technique versus complex resampling processors?
- 5) How real is the desire to provide deterministic data acquisition through sensor pointing? Are the cost savings of deterministic data acquisition greater than the cost of producing the curved focal plane array?

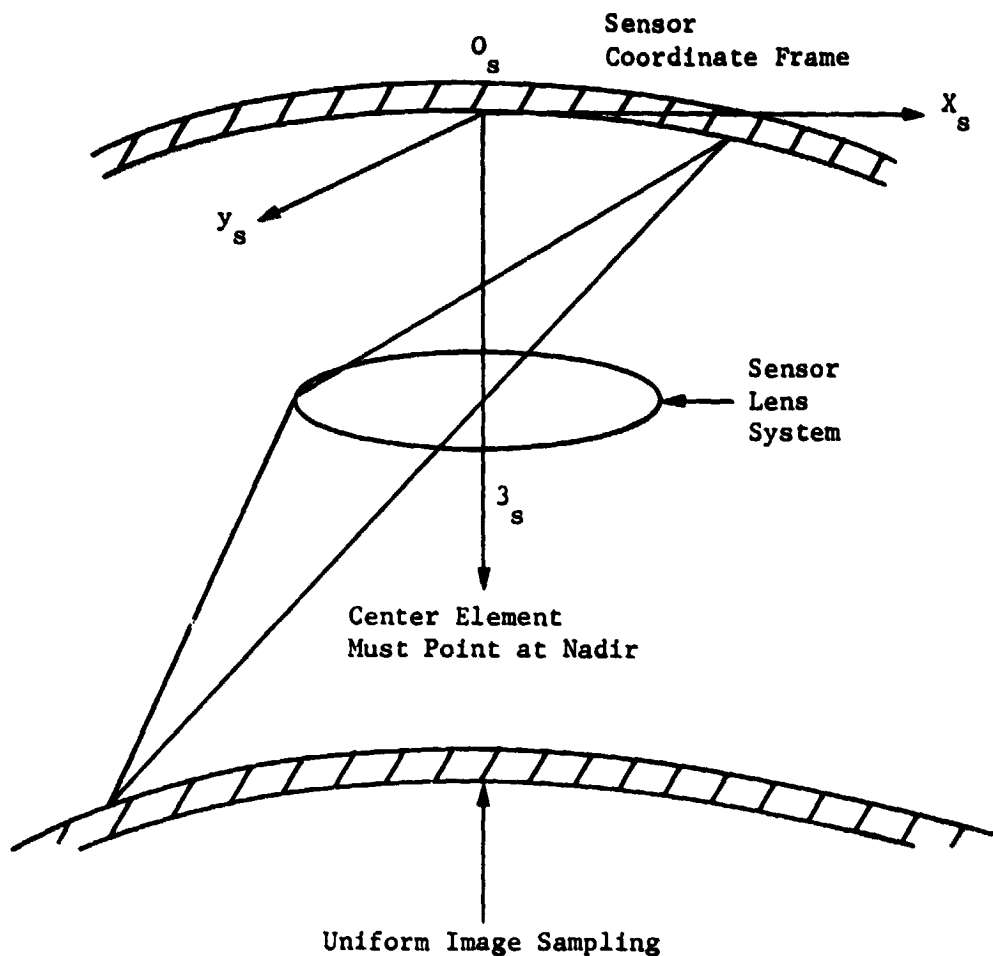


Figure II-13 Curved Focal Plane Array

The answers to these questions will form the basis for selecting the system architecture to be used on the next generation of remote sensing vehicles. However, even if a curved focal plane is not chosen, the separation of the science instrument from the motion of the spacecraft through the use of a pointing mount will reduce the distortions and simplify the process of image correction while providing deterministic data acquisition.

III. GCP NAVIGATION SYSTEM SIMULATION

The GCPSIM program has been configured to provide scientific simulations to predict the performance of the GCP detection system. Two types of simulation modes have been incorporated. The first mode simulates the spacecraft the spacecraft environment and all measurements. The second mode extracts ground control point measurements directly from Landsat imagery.

The program was set up to provide extensive error analysis rather than simply a covariance analysis. Although covariance analysis provides a great deal of information, in many cases this information can be inaccurate or misleading. For example, there are cases where the covariance matrix converges over a period of time while the actual state estimate diverges from the true state. For this reason the simulation models the true state of the vehicle using the dynamic equations of motion. By propagating the true vehicle state, one can perform a covariance analysis and also generate the actual state error and measurement residuals. The two modes of operation are described in the following sections. A thorough description of the software is contained in Appendix A.

A. SIMULATED GCP MEASUREMENT

GCPSIM was designed to provide the ability to analyze the effect of various measurement sequences. This was especially important when studying the effect of GCP spacing, missed GCP sightings, and the expected accuracy after traversing a large body of water. The measurement sequencer designed for GCPSIM allows any mixture of GCP, global positioning system or star tracker measurements and time delays (periods during which no measurements were made) of any length. The sequencer determines the type of measurement and the time at which the measurement should be made. The true vehicle position state is then propagated forward to this time by integrating the nonlinear equations of motion with some additional process noise to account for modeling errors. The attitude state is propagated by looking up the body rates in an attitude profile table and integrating these rates.

The true vehicle state is used along with a measurement model to generate an ideal measurement vector. The ideal measurement is then corrupted with noise, bias, and misalignment terms and compensated for knowledge of these values. This allows a careful analysis of the effect of misalignment on the state solution. It is important to understand the effect of bias and misalignment between the landmark tracker and body axis because this is the largest unknown factor contributing to a pointing error. It is possible to provide frequent inflight calibration for these misalignment errors, but it is difficult to model, for any length of time, the various processes that cause the misalignment. For example, thermal gradients across the vehicle and vibrational modes within the flexible structure are complex functions of such things as

structural design, sun angle, physical properties of the material and many other factors. These processes are the most difficult and least understood of all engineering problems.

The compensated measurements are used as inputs into an extended square root Kalman filter, which estimates the true vehicle state. The extended filter propagates the estimated navigation state, the state transition matrix and the process noise array between measurements by integrating the various differential equations using a fourth order Runge Kutta Gill process. It is anticipated that a linear propagation would be satisfactory for 5-second and GPS update intervals. However, since the GPS update interval was one of the parameters investigated, the more general approach of state propagation through integration was used. The estimated attitude state is propagated by a gyro model that corrupts the output with gyro drift, noise, nonorthogonality, scale factor and misalignment. The gyro output is compensated in a similar fashion to the measurement model.

The estimated state is used to form an estimated measurement, which in turn is subtracted from the true measurement to obtain a residual. This measurement residual and calculated Kalman gain are used to update the state estimate, which can be compared with the true state to yield the error. The entire process continues until the spacecraft is propagated forward to the run stop time.

The remainder of this section describes each of these processes in more detail, and Appendix A provides a detailed software description of the program.

1. Measurement Models

This section summarizes the mathematical measurement models used in the simulation of an onboard GCP detection system. The types of measurements used include GPS position and velocity, star tracker sightings, and landmark tracker sightings. The design philosophy behind the measurement models is that the actual vehicle state is used with a geometry model to yield an ideal measurement vector. This ideal vector is then corrupted with bias, noise, and misalignment to provide the actual sensor output. The sensor output is then compensated for some estimate of the error terms. This compensated measurement then is used by the filter to estimate the vehicle state. The benefit this design approach provides is that it enables a detailed analysis of sensitivity to misalignments and compensation ability. It is also expected that the severe requirements associated with onboard image correction will require the onboard estimate of misalignment terms such as those between the MSS and body coordinates. With this approach it will not be difficult to modify the filter to solve for these terms. The remainder of this section has been separated into subsections corresponding to each measurement mode.

- a. Landmark Tracker Measurement Model Development - The landmark location on the surface of the earth in local landmark coordinates will be a function of the altitude (A_L) above the earth's mean radius

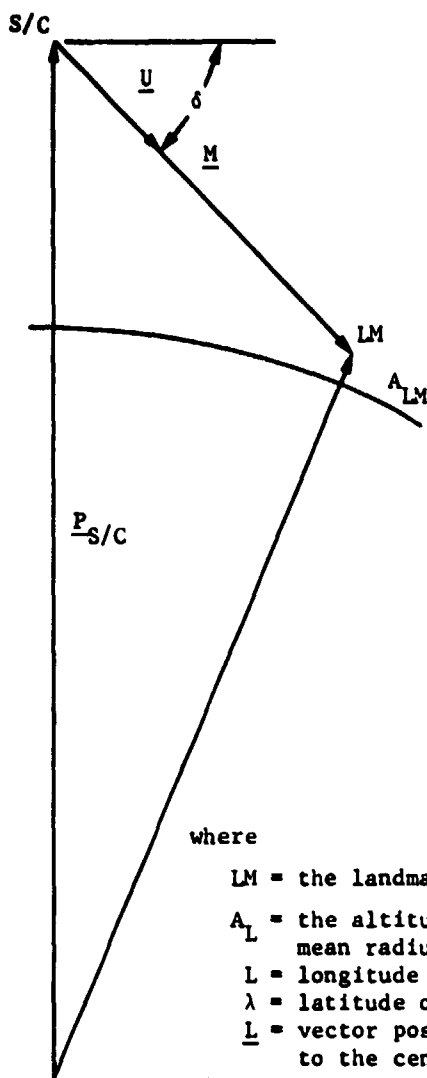
$$\underline{L}_L = \begin{bmatrix} A_L \\ 0 \\ 0 \end{bmatrix}$$

However, in earth-fixed coordinates, the landmark will have the earth's mean radius (r_E) added to the altitude. Using the angular transformation from local landmark to earth-fixed coordinates produces

$$\begin{aligned} \underline{L}_E &= E^T L_L = E^T \begin{bmatrix} r_E + A_L \\ 0 \\ 0 \end{bmatrix} \\ &= (r_E + A_L) \begin{bmatrix} CLC\lambda & -SL & -CLS\lambda \\ SLC\lambda & CL & -SLS\lambda \\ S\lambda & 0 & C\lambda \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &= (r_E + A_L) \begin{bmatrix} CLC\lambda \\ SLC\lambda \\ S\lambda \end{bmatrix} \end{aligned}$$

As shown in Figure III-1, the position vector of the spacecraft ($P_{S/C}$), when subtracted from the landmark position in some coordinate frame, will provide the measurement vector (M)

$$\underline{M}_I = (I^T E \underline{L}_E) - \underline{P}_{S/C_I}$$



where

- LM = the landmark being used = $f(L, \lambda, A_L)$,
- A_L = the altitude of the landmark above the mean radius of the earth,
- L = longitude of the landmark,
- λ = latitude of the landmark,
- \underline{L} = vector position of the landmark relative to the center of the earth,
- $\underline{P}_{S/C}$ = vector position of the spacecraft relative to the center of the earth,
- $A_{S/C}$ = altitude of the spacecraft above the mean radius of the earth,
- \underline{M} = measurement vector from the spacecraft to the landmark,
- \underline{U} = unit vector along \underline{M} ,
- \underline{H} = the landmark tracker horizontal plane angular deflection from the boresight axis,
- \underline{V} = the landmark tracker vertical plane angular deflection from the boresight axis.

Figure III-1
Landmark Tracker Geometry -
Sighting Plan

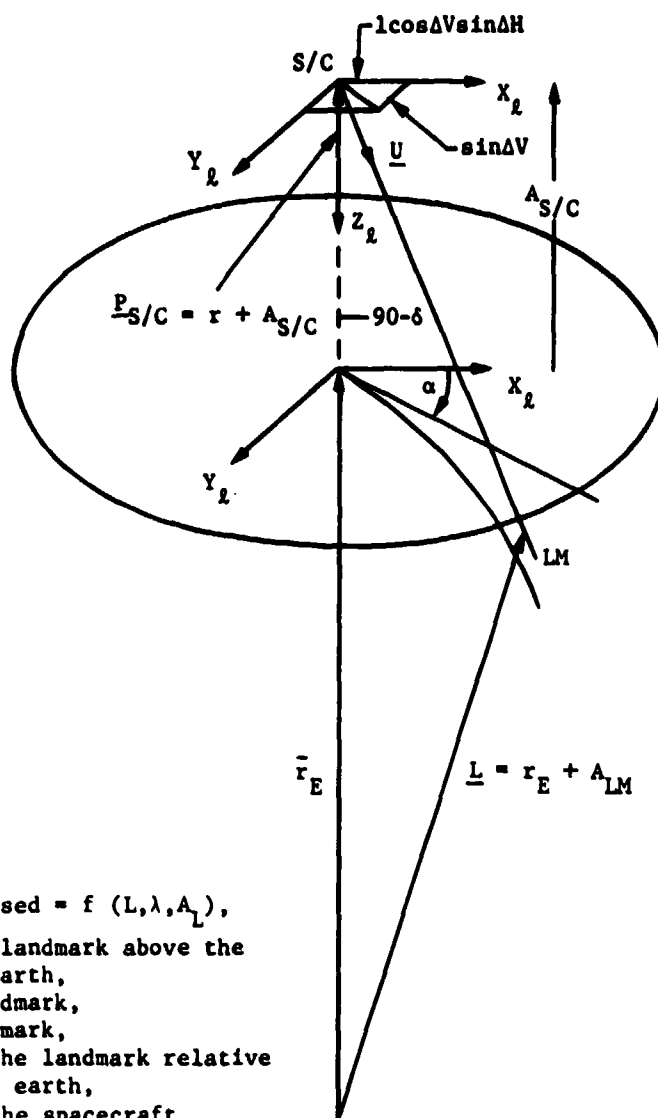


Figure III-2
Landmark Tracker Geometry -
General

Accounting for hardware misalignments, the same measurement vector in landmark tracker coordinate is

$$\begin{aligned}\underline{M}_I &= {}^I T_I ({}^I T_E \underline{L}_E) - \underline{P}_S / C_I \\ &= {}^I T_E \underline{L}_E - {}^I T_I \underline{P}_S / C_I.\end{aligned}$$

From examination of Figure III-2, the unit measurement vector in landmark tracker coordinates is

$$\underline{U}_\ell = \frac{\underline{M}_\ell}{|\underline{M}_\ell|} = \begin{bmatrix} U_{\ell x} \\ U_{\ell y} \\ U_{\ell z} \end{bmatrix} = \begin{bmatrix} \sin \Delta V \\ \cos \Delta V \sin \Delta H \\ \cos \Delta V \cos \Delta H \end{bmatrix}.$$

However, the tracker instrument has no sensitivity to projections along its boresight axis. Therefore the tracker response to the unit vector \underline{U}_ℓ will be

$$\underline{U}_\ell = \begin{bmatrix} U_{\ell x} \\ U_{\ell y} \end{bmatrix} = \begin{bmatrix} \sin \Delta V \\ \cos \Delta V \sin \Delta H \end{bmatrix}$$

producing a ΔH and ΔV as shown in Figure III-2 as sensor outputs. Since the sensor output will be corrupted by bias and noise, the sensed measurement will be

$$\underline{Z}_\ell = \begin{bmatrix} \Delta H_s \\ \Delta V_s \end{bmatrix} = \begin{bmatrix} \Delta H + b_H + v_H \\ \Delta V + b_V + v_V \end{bmatrix}$$

where

b_H, b_V = component landmark tracker bias,
 v_H, v_V = component landmark tracker zero mean random noise, $N(0, \sigma^2)$.

The component biases and standard deviations (σ) are user selectable.

The landmark tracker measurement may be compensated for knowledge of instrument bias. The bias knowledge may be a priori or through estimation. The compensated sensor output will be

$$\hat{\underline{Z}}_\ell = \begin{bmatrix} \Delta H_c \\ \Delta V_c \end{bmatrix} = \begin{bmatrix} \Delta H_s - \hat{b}_H \\ \Delta V_s - \hat{b}_V \end{bmatrix}.$$

- b. Global Positioning System (GPS) Model Development - The function of the GPS received is to yield an estimate of the user's position and velocity in inertial space based on signals from four GPS satellites. The detail associated with the modeling the navigation signals from every GPS satellite is unnecessary for this simulation. A sufficient model simply adds measurement noise in the form of a position-velocity uncertainty to the user's true inertial position-velocity vector. This uncertainty is assumed to be made up of component uncertainties along the inertial coordinate axes. These components are assumed to be normally distributed with zero mean and standard deviation σ_i , $i = X_I, Y_I, Z_I, \dot{X}_I, \dot{Y}_I, \dot{Z}_I$.

The GPS measurement vector is therefore

$$\underline{Z}_G = \begin{bmatrix} \underline{X} \\ \dots \\ \dot{\underline{X}} \end{bmatrix} + \underline{V} + \underline{B}(t)$$

where

\underline{Z}_G = the GPS measurement vector (6 x 1),

\underline{X} = the user's inertial position vector (3 x 1),

$\dot{\underline{X}}$ = the user's inertial velocity vector (3 x 1),

\underline{V} = The position-velocity uncertainty vector $\{V_X, V_{\dot{X}}\}^T$, the components of which are $N_i ((0, \sigma_i^2))$ (6 x 1),

$\underline{B}(t)$ = The position-velocity bias vector at time t.

When compensation is made for bias knowledge by subtracting that knowledge, the resulting compensated measurement is

$$\hat{\underline{Z}}_G = \underline{Z}_G - \hat{\underline{B}}$$

where

$(\hat{\underline{B}})$ = The position-velocity bias knowledge (6 x 1).

- c. Star Tracker Measurement Model Development - When obtaining star tracker measurements, it is important to identify when the field of view is blocked by a major body such as the sun, moon, or earth. The following paragraph describes this process.

Occultation - Occultation occurs when any major body comes between the spacecraft and the target star or if a sufficiently bright object falls within the field of view of the tracker to render the device inoperable. For the simplified star tracker model, occultation will be defined to have occurred if the tracker boresight axis touches any part of one of three major bodies--sun, earth, or moon.

Accounting for hardware misalignments, the same measurement vector in landmark tracker coordinate is

$$\begin{aligned}\underline{M}_1 &= {}^1T_I ({}^I T_E \underline{L}_E) - \underline{P}_S/C_I \\ &= {}^1T_E \underline{L}_E - {}^1T_I \underline{P}_S/C_I.\end{aligned}$$

From examination of Figure III-2, the unit measurement vector in landmark tracker coordinates is

$$\underline{U}_L = \frac{\underline{M}_L}{|\underline{M}_L|} = \begin{bmatrix} U_{Lx} \\ U_{Ly} \\ U_{Lz} \end{bmatrix} = \begin{bmatrix} \sin \Delta V \\ \cos \Delta V \sin \Delta H \\ \cos \Delta V \cos \Delta H \end{bmatrix}.$$

However, the tracker instrument has no sensitivity to projections along its boresight axis. Therefore the tracker response to the unit vector \underline{U}_L will be

$$\underline{U}_L = \begin{bmatrix} U_{Lx} \\ U_{Ly} \end{bmatrix} = \begin{bmatrix} \sin \Delta V \\ \cos \Delta V \sin \Delta H \end{bmatrix}$$

producing a ΔH and ΔV as shown in Figure III-2 as sensor outputs. Since the sensor output will be corrupted by bias and noise, the sensed measurement will be

$$\underline{Z}_L = \begin{bmatrix} \Delta H_s \\ \Delta V_s \end{bmatrix} = \begin{bmatrix} \Delta H + b_H + V_H \\ \Delta V + b_V + V_V \end{bmatrix}$$

where

b_H, b_V = component landmark tracker bias,
 V_H, V_V = component landmark tracker zero mean random noise, $N(0, \sigma^2)$.

The component biases and standard deviations (σ) are user selectable.

The landmark tracker measurement may be compensated for knowledge of instrument bias. The bias knowledge may be a priori or through estimation. The compensated sensor output will be

$$\hat{\underline{Z}}_L = \begin{bmatrix} \Delta H_c \\ \Delta V_c \end{bmatrix} = \begin{bmatrix} \Delta H_s - \hat{b}_H \\ \Delta V_s - \hat{b}_V \end{bmatrix}.$$

- b. Global Positioning System (GPS) Model Development - The function of the GPS received is to yield an estimate of the user's position and velocity in inertial space based on signals from four GPS satellites. The detail associated with the modeling the navigation signals from every GPS satellite is unnecessary for this simulation. A sufficient model simply adds measurement noise in the form of a position-velocity uncertainty to the user's true inertial position-velocity vector. This uncertainty is assumed to be made up of component uncertainties along the inertial coordinate axes. These components are assumed to be normally distributed with zero mean and standard deviation σ_i , $i = X_1, Z_1, X_1, Y_1, Z_1$.

The GPS measurement vector is therefore

$$\underline{Z}_G = \begin{bmatrix} \underline{X} \\ \dots \\ \dot{\underline{X}} \end{bmatrix} + \underline{V} + \underline{B}(t)$$

where

\underline{Z}_G = the GPS measurement vector (6 x 1),

\underline{X} = the user's inertial position vector (3 x 1),

$\dot{\underline{X}}$ = the user's inertial velocity vector (3 x 1),

\underline{V} = The position-velocity uncertainty vector $\{V_X, V_X\}^T$, the components of which are N_i $((0, \sigma_i^2))$ (6 x 1),

$\underline{B}(t)$ = The position-velocity bias vector at time t.

When compensation is made for bias knowledge by subtracting that knowledge, the resulting compensated measurement is

$$\hat{\underline{Z}}_g = \underline{Z}_g - \hat{\underline{B}}$$

where

$(\hat{\underline{B}})$ = The position-velocity bias knowledge (6 x 1).

- c. Star Tracker Measurement Model Development - When obtaining star tracker measurements, it is important to identify when the field of view is blocked by a major body such as the sun, moon, or earth. The following paragraph describes this process.

Occultation - Occultation occurs when any major body comes between the spacecraft and the target star or if a sufficiently bright object falls within the field of view of the tracker to render the device inoperable. For the simplified star tracker model, occultation will be defined to have occurred if the tracker boresight axis touches any part of one of three major bodies--sun, earth, or moon.

Considering Figure III-1

$$\beta = \sin^{-1} \frac{r_1}{|R_1|}, \quad |R_1| = \sqrt{(X_1 - X_{sc})^2 + (Y_1 - Y_{sc})^2 + (Z_1 - Z_{sc})^2}.$$

Since

$$\underline{U}_s \cdot \underline{R}_1 = \underline{U}_s^T \underline{R}_1 = |\underline{U}_s| |\underline{R}_1| \cos \alpha,$$

$$\alpha = \cos^{-1} \frac{\underline{U}_s^T \underline{R}_1}{|\underline{U}_s| |\underline{R}_1|}$$

where \underline{R}_i and \underline{U}_s are known in the same coordinate system, and occultation is defined to have occurred if $\alpha \leq \beta$.

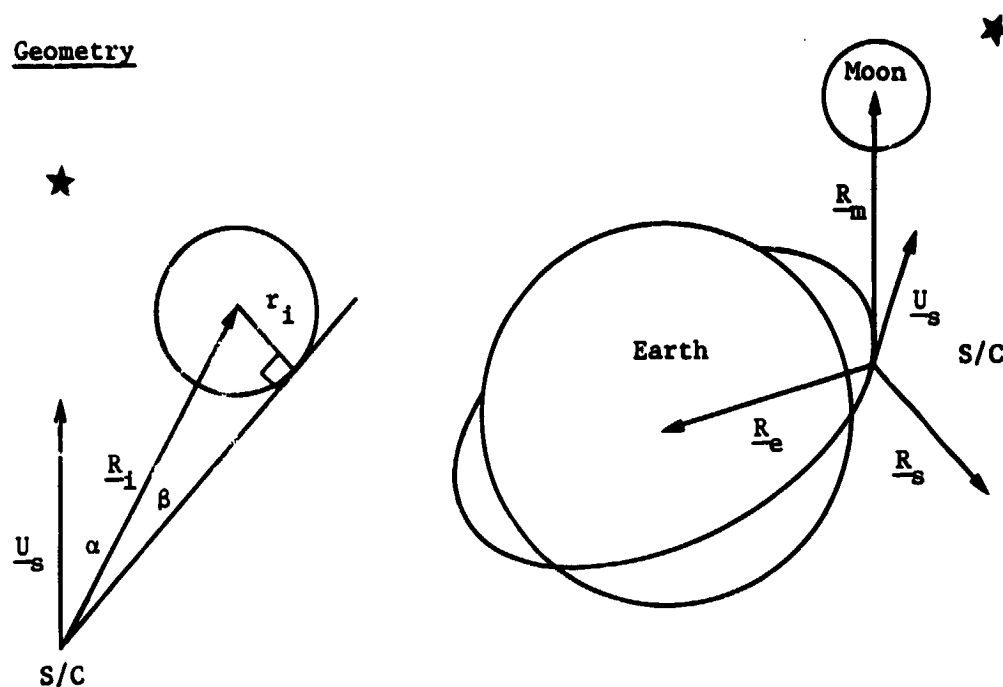
Under these circumstances, a star tracker update is not available from the occulted tracker.

Nonoccultation Star Tracker Model (Simplified) - If occultation has not occurred, a reference star will be defined to have been sighted on the star tracker boresight axis. In this simplified case, this will be modeled as a unit vector along the boresight axis. A bias and zero mean random noise will be added to the measurement vector. Therefore the corrupted unit vector in the i th star tracker coordinates will be

$$\underline{M}_{si} = \underline{U}_{si} + \underline{b}_{si} + \underline{v}_{si} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ 0 \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix}.$$

The bias elements b_x and b_y will be user selectable. The noise elements v_x and v_y will be zero mean normal variables $N(0, \sigma_v^2)$, where σ_{v_x} and σ_{v_y} will be user selectable.

Geometry



Occulting Geometry

Simulation Geometry

\underline{R}_e , \underline{R}_m , \underline{R}_s = line-of-sight vectors from the spacecraft (S/C) to the center of the earth, moon, and sun, respectively,

\underline{R}_i = general radius vector to the i^{th} celestial object,

r_i = the radius of the i^{th} planet,

\underline{U}_s = the unit vector to the target star

= The unit vector along the star tracker boresight axis (simplified model),

= $\{0,0,1\}^T$ in tracker coordinates (simplified model)

β = half-cone angle from the spacecraft to the i^{th} major body center and body limb,

α = angle from the spacecraft to the i^{th} major body center and the target star.

Figure III-3 Star Tracker Measurement Model Development

Since the star tracker has no sensitivity to projections along its boresight, only components perpendicular to the boresight axis in the tracker coordinates will be sensed. The sensed measurement for the i th star tracker will be

$$\bar{z}_{s_i} = \begin{bmatrix} \Delta H_{s_i} \\ \Delta V_{s_i} \end{bmatrix} = \begin{bmatrix} bx + Vx \\ by + Vy \end{bmatrix}.$$

The sensed measurement may be compensated for known biases. This bias is assumed to be known a priori and not estimated at this time. The compensated measurement vector is therefore

$$\hat{z}_{s_i} = \begin{bmatrix} \hat{\Delta H}_{s_i} \\ \hat{\Delta V}_{s_i} \end{bmatrix} = \begin{bmatrix} \Delta H_{s_i} - \hat{b}_H \\ \Delta V_{s_i} - \hat{b}_V \end{bmatrix}.$$

Gyro Model - The general function of the gyro model is to generate and process the angular changes sensed by the strapdown gyros at evenly spaced time intervals determined by one scan line of the science sensor.

Processing consists of correction for gyro-considered parameters--drift, nonorthogonality, scale factor, and misalignment. The compensated gyro output is used to update the quaternion so the attitude of the spacecraft can be known as accurately as possible at all times. It is then used to propagate the attitude state transition matrix, which is used in turn to propagate the covariance matrix.

The major input to the gyro model is the angular changes of the spacecraft during each sample period. The length of this period defaults to 0.1 second. The input data consist of three variables that correspond to the angular changes about the three orthogonal body axes.

Other input data required by the gyro model are the initial value of the attitude quaternion and the time at which it is valid (the last attitude reference measurement time). The attitude quaternion, which is propagated by the gyro model, is later updated by the Kalman filter when an attitude reference measurement is processed.

The gyro model first simulates the gyro output for a specified interval. This is a three-step process in which the true attitude rate is obtained from an attitude profile table. These rates are transformed into individual gyro coordinates through

$$\Delta \theta_G = G_G^T \cdot G_B^T \Delta \theta_B$$

where

$\Delta\theta_I$ = change in spacecraft attitude in body coordinates,

G^{-T}_B = transformation from body to gyro coordinates accounting for misalignment of gyro package (the inverse of this transformation is expanded below),

$G^{T}_{G'}$ = transformation from gyro package coordinates to individual gyro coordinates accounting for nonorthogonality and scale factor, (the inverse of this transformation is expanded below),

$\Delta\theta_G$ = change in angle sensed by individual gyros.

The change in angle sensed by the individual gyros is corrupted with gyro drift and random noise to yield the modeled gyro output

$$\Delta\theta_M = (\Delta\theta_G + \beta + \eta) \times \Delta T$$

where

$\Delta\theta_M$ = gyro output,

β = gyro bias,

η = gyro noise

ΔT = sampling interval.

The gyro output is processed to eliminate gyro systematic errors (process noise). The systematic errors include the constant drift rate, the scale factor, the nonorthogonality of the gyro input axes, and the misalignment of the ideal gyro input axes with respect to the spacecraft axis system. These compensated angular changes are calculated by the equation

$$\begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{bmatrix} = B^T_G \cdot G^{T}_{G'} \begin{bmatrix} \Delta\theta_{1m} + W_{Dx} \cdot \Delta t \\ \Delta\theta_{2m} + W_{Dy} \cdot \Delta t \\ \Delta\theta_{3m} + W_{Dz} \cdot \Delta t \end{bmatrix}$$

where

B^T_G = gyro misalignment transformation matrix,

$G^{T}_{G'}$ = gyro scale factor and nonorthogonality transition matrix,

$\Delta\theta_1, \Delta\theta_2, \Delta\theta_3$ = compensated gyro data (radians),

$\Delta\theta_{1m}, \Delta\theta_{2m}, \Delta\theta_{3m}$ = input data from gyros (radians),

w_{Dx}, w_{Dy}, w_{Dz} = gyro constant drift rate (rad/sec),

Δt = gyro measurement interval (s).

The matrix $B^T G$ has the following components:

$$G^T G' = \begin{bmatrix} S_x & 0 & 0 \\ -S_x \cdot \delta_1 & S_y & 0 \\ -S_x \cdot \delta_2 & -S_y \cdot \delta_3 & S_z \end{bmatrix}$$

where

S_x, S_y, S_z = gyro scale factors (radians/count)

$\delta_1, \delta_2, \delta_3$ = gyro misalignment angles (radians)

Notice that $G^T G'$ is the inverse of the transformation used to model δ the individual gyro outputs. However, different values of S and δ are used for modeling and compensation for realistic modeling of system uncertainties.

The matrix $B^T G$ has the following components

$$B^T G = \begin{bmatrix} C\Delta_3 \cdot C\Delta_2 & S\Delta_3 \cdot C\Delta_2 & -S\Delta_2 \\ S\Delta_1 \cdot S\Delta_2 \cdot S\Delta_3 - C\Delta_1 \cdot S\Delta_3 & S\Delta_1 \cdot S\Delta_2 \cdot S\Delta_3 + C\Delta_1 \cdot C\Delta_3 & C\Delta_2 \cdot S\Delta_1 \\ C\Delta_1 \cdot S\Delta_2 \cdot C\Delta_3 + S\Delta_1 \cdot S\Delta_3 & C\Delta_1 \cdot S\Delta_2 \cdot S\Delta_3 - S\Delta_1 \cdot C\Delta_3 & C\Delta_2 \cdot C\Delta_1 \end{bmatrix}$$

where the abbreviations C and S are used for cosine and sine, respectively, and $\Delta_3, \Delta_2, \Delta_1$ are Euler angles representing rotations about the z axis, y axis, and x axis, in that order. Again, $B^T G$ is the inverse of the transformation used to model gyro output but the values of Δ_3, Δ_2 , and Δ_1 are different.

The compensated gyro data are then used to compute the differential of the quaternion that is used to propagate the quaternion. The quaternion is used to update the attitude transition matrix. The net result of this process is an update in the attitude estimate and a new attitude transition matrix. The updated quaternion may be found through

$$\begin{bmatrix} e'_0 \\ e'_1 \\ e'_2 \\ e'_3 \end{bmatrix} = \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} + 1/2 \begin{bmatrix} -e_1 & -e_3 & -\Delta\theta_3 \\ e_0 & -e_2 & e_2 \\ e_3 & e_0 & -e_1 \\ -e_2 & e_1 & e_0 \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{bmatrix}$$

where

e_0, e_1, e_2, e_3 = old quaternion,

e'_0, e'_1, e'_2, e'_3 = updated quaternion,

$\Delta\theta_1, \Delta\theta_2, \Delta\theta_3$ = compensated gyro output.

An overview of the measurement models is illustrated in Figure III-4.

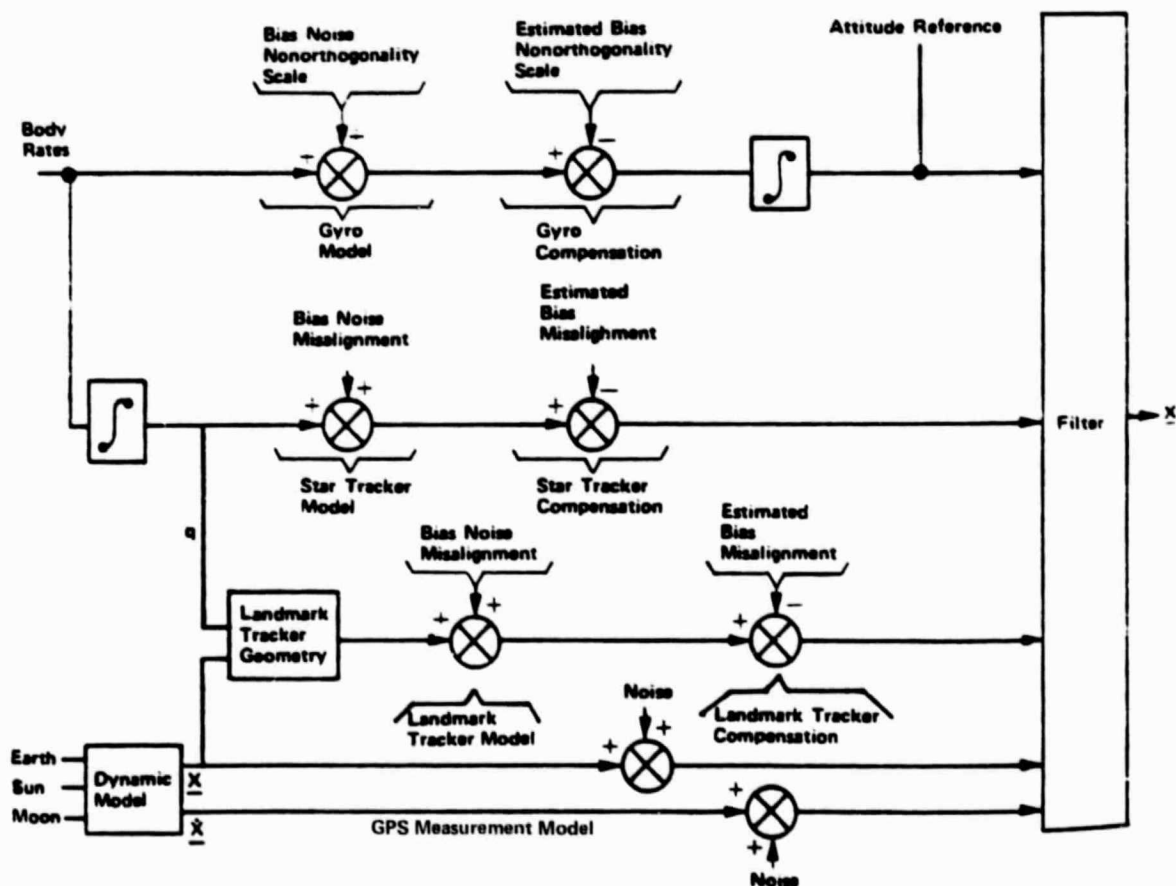


Figure III-4 Overview of Measurement Models

2. Dynamics Model and Navigation State Integrator

a. Dynamics Model - The dynamics model calculates the derivative of the spacecraft navigational state, which will be integrated to produce the navigational state vector. This is done in part by calculating the total acceleration of the spacecraft due to solar pressure and gravitation effects of the sun, moon, and earth, including fourth zonal harmonic terms.

The sequential process for determining the forces acting on the spacecraft is:

- 1) Calculate Julian data
- 2) Calculate solar and lunar positions
- 3) Calculate solar pressure acceleration
- 4) Calculate gravitational acceleration due to sun and moon
- 5) Calculate gravitational acceleration due to earth using four zonal harmonic terms
- 6) Calculate total acceleration

Algorithm and Process - The position of the spacecraft is calculated by solving three simultaneous, second-order differential equations

$$\ddot{X}_1 = -X_1 \cdot \frac{\mu}{R^3} + g_1(t, X) + a_1(t, X)$$

$$\ddot{X}_2 = -X_2 \cdot \frac{\mu}{R^3} + g_2(t, X) + a_2(t, X)$$

$$\ddot{X}_3 = -X_3 \cdot \frac{\mu}{R^3} + g_3(t, X) + a_3(t, X)$$

where

$$\underline{X} = [X_1, X_2, X_3]^T$$

μ = earth gravitational constant ($3.985491204E + 14 \text{ m}^3/\text{s}^2$),

$$R = (X_1^2 + X_2^2 + X_3^2)^{1/2},$$

X_1, X_2, X_3 = coordinates of spacecraft,

g_1, g_2, g_3 = accelerations caused by zonal harmonics of earth gravity,

a_1, a_2, a_3 = solar radiation pressure perturbations, sun and moon gravity.

The equations for computation of the zonal harmonic accelerations are

$$g_1 = -\frac{\mu X_1}{R^3} F_1$$

$$g_2 = -\frac{\mu X_2}{R^3} F_1$$

$$g_3 = -\frac{\mu}{R^2} \left[\frac{X_3}{R} F_1 - F_2 \right]$$

where

$$F_1 = \frac{R_e^2}{R} \cdot f_2 J_2 + \frac{R_e^3}{R} \cdot f_3 J_3 + \frac{R_e^4}{R} \cdot f_4 J_4,$$

$$F_2 = \frac{R_e^2}{R} \cdot f_1 J_2 + \frac{R_e^3}{R} \cdot f_2 J_3 + \frac{R_e^4}{R} \cdot f_3 J_4,$$

and

$$f_1 = -3 \frac{X_3}{R},$$

$$f_2 = -7.5 \left(\frac{X_3}{R}\right)^2 + 1.5,$$

$$f_3 = -17.5 \left(\frac{X_3}{R}\right)^3 + 7.5 \left(\frac{X_3}{R}\right),$$

$$f_4 = -39.375 \left(\frac{X_3}{R}\right)^4 + 26.25 \left(\frac{X_3}{R}\right)^2 - 1.875,$$

$$J_2 = 1082.7E-6 \text{ (Harmonic term in earth gravity model),}$$

$$J_3 = -2.56E-6 \text{ (Harmonic term in earth gravity model),}$$

$$J_4 = -1.58E-6 \text{ (Harmonic term in earth gravity model).}$$

b. Navigation State Integrator - The position state is advanced in time by numerical integration of the equations of motion. The second-order equations of motion are composed of the external forces acting on the spacecraft. The external forces consist of geopotential, lunar and solar gravitation, and radiation pressure. The integration routine is used to propagate both the actual state and the estimated state between measurement time.

Algorithm and Process - A study showed that the Runge Kutta Gill (RKG) fourth order numerical integration method is optimal for this application. It is self-starting, handles variable step sizes, and is sufficiently accurate. The RKG method for numerically integrating differential equations is described here.

The change in the value of the function during the computing interval is calculated by

$$\Delta y = \frac{1}{6} \left[k_1 + 2(1-\mu)k_2 + 2(1+\mu)k_3 + k_4 \right]$$

where

$$k_1 = h \cdot f(t_n, y_n), \mu = \sqrt{2}/2,$$

$$k_2 = h \cdot f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1),$$

$$k_3 = h \cdot f(t_n + \frac{1}{2}h, y_n + (-\frac{1}{2} + \mu)k_1 + (1 - \mu)k_2),$$

$$k_4 = h \cdot f(t_n + h, y_n - \mu k_2 + (1 + \mu)k_3),$$

h = computing interval (seconds),

t_n = time of beginning of computing interval (seconds),

y_n = value of function at beginning of computing interval.

The derivative function f shall be evaluated four times to calculate the change in the function being integrated during the computing interval.

At first appearance, the software algorithm shown in Figure III-5 has no relation to the mathematical description of the integrator. The following is intended to show that the two are indeed identical.

Initial Entry	
T	F
Null 0 Matrix	Null
Compute Position Dynamics	
K = 1 or 3	
T	F
T = T + DT/2	Null
TP = AA(K) x [ZD(I) - BB(K) x O(I)]	
W = Z(I)	
Z(I) = Z(I) + TP x DT	
TP = [Z(I) - W]/DT	
O(I) = O(I) + 3 x TP - CC(K) x ZD(I)	
where AA - 0.5, 1 - 2/2, 1 + 2/2, 0.5/3; BB - 2,1,1,2; CC - 0.5, 1 - 2/2, 1 + 2/2, 0.5	
Do Until I = 6	
Do Until K = 4	

Figure III-5 Software Algorithm

Proceeding through the outer "do" loop for $K = 1$ to 4, we find the following results

$$k = 1 \quad f = f(t_n, y_n)$$

$$t_1 = \frac{1}{2}f$$

$$y_1 = y_0 + \frac{1}{2}dtf$$

$$q_1 = f$$

$$k = 2 \quad f' = f(t_n + \frac{dt}{2}, y_0 + \frac{dt}{2}f)$$

$$t_2 = (1 - \frac{\sqrt{2}}{2})(f' - f)$$

$$y_2 = y_0 + \frac{dt}{2}f + dt(1 - \frac{\sqrt{2}}{2})(f' - f)$$

$$= y_0 + dt(-\frac{1}{2} + \frac{\sqrt{2}}{2})f + dt(1 - \frac{\sqrt{2}}{2})f'$$

$$q_2 = f + 3((1 - \frac{\sqrt{2}}{2})(f' - f)) - (1 - \frac{\sqrt{2}}{2})f'$$

$$= f - 3(1 - \frac{\sqrt{2}}{2})f + 2(1 - \frac{\sqrt{2}}{2})f'$$

$$k = 3 \quad f'' = f(t_n + \frac{dt}{2}, y + dt(-\frac{1}{2} + \frac{\sqrt{2}}{2})f + dt(1 - \frac{\sqrt{2}}{2})f')$$

$$t_3 = (1 + \frac{\sqrt{2}}{2})(f'' - f + 3(1 - \frac{\sqrt{2}}{2})f - 2(1 - \frac{\sqrt{2}}{2})f')$$

$$y_3 = y_0 + dt(-\frac{1}{2} + \frac{\sqrt{2}}{2})f + dt(1 - \frac{\sqrt{2}}{2})f' + dt(1 + \frac{\sqrt{2}}{2})(f'' - f + 3(1 - \frac{\sqrt{2}}{2})f - 2(1 - \frac{\sqrt{2}}{2})f') = y_0 - dt\frac{\sqrt{2}}{2}f' + dt(1 + \frac{\sqrt{2}}{2})f''$$

$$q_3 = f - 3(1 - \frac{\sqrt{2}}{2})f + 2(1 - \frac{\sqrt{2}}{2})f' + 3((1 + \frac{\sqrt{2}}{2})(f'' - f + 3(1 - \frac{\sqrt{2}}{2})f - 2(1 - \frac{\sqrt{2}}{2})f')) - (1 + \frac{\sqrt{2}}{2})f''$$

$$= -\frac{1}{2}f - (1 + \frac{\sqrt{2}}{2})f' + 2(1 + \frac{\sqrt{2}}{2})f''$$

$$k = 4 \quad f''' = f(t_n + dt, y_0 - dt\frac{\sqrt{2}}{2}f' + dt(1 + \frac{\sqrt{2}}{2})f'')$$

$$t_4 = 1/6(f''' - 2 - \frac{1}{2}f - (1 + \frac{\sqrt{2}}{2})f' + 2(1 + \frac{\sqrt{2}}{2})f'')$$

$$= 1/6(f''' + f + 2(1 + \frac{\sqrt{2}}{2})f' - 4(1 + \frac{\sqrt{2}}{2})f'')$$

$$y_4 = y_0 - dt\frac{\sqrt{2}}{2}f' + dt(1 + \frac{\sqrt{2}}{2})f'' + 1/6(dt f + 2dt(1 + \frac{\sqrt{2}}{2})f' - 4dt(1 + \frac{\sqrt{2}}{2})f'' + dt f''')$$

$$= y_0 + 1/6(dt f + 2dt(1 - \frac{\sqrt{2}}{2})f' + 2dt(1 + \frac{\sqrt{2}}{2})f'' + dt f''').$$

Making the substitutions

$$k_1 = dtf$$

$$k_2 = dtf'$$

$$k_3 = dtf''$$

$$k_4 = dtf'''$$

$$k = \frac{\sqrt{2}}{2}$$

yields

$$y - y_0 = \Delta y = 1/6\{k_1 + 2(1 - k)k_2 + 2(1 + k)k_3 + k_4\},$$

which is identical to the original algorithm.

3. The Extended Carlson Square Root Filter Mechanization

The Carlson square root filter is an extension of the conventional Kalman filter where the optimal gain, state and covariance updates are determined using a matrix that is the square root of the conventional covariance matrix. The advantages of this mechanization are higher accuracy for a given machine precision and a guaranteed positive covariance. The extended filter is necessary due to the nonlinear nature of the process. The extended filter propagates the state, state transition, and process noise covariance by integration of differential equations rather than linear propagation over the appropriate intervals.

The model equations used for the filter mechanization are the state differential equations

$$\underline{X} = f(\underline{X}(t), t) + \underline{W}(t), \quad \underline{W}(t) = N(0, Q(t))$$

where

$$Q(t) = E(\underline{W}(t)\underline{W}(t)^T),$$

E = expectation operator,

$\underline{W}(t)$ = normal random variable process noise with zero mean and covariance, $Q(t)$,

f = a nonlinear function of $\underline{X}(t)$ and t,

the measurement equation

$$\underline{Z}_k = \underline{h}_k(\underline{X}(t_k)) + \underline{V}_k, \quad \underline{V}_k = N(0, R_k)$$

where

R_k = the covariance of V_k

V_k = normal random variable measurement noise with zero mean and covariance, R_k ,

and the measurement estimate equation

$$\hat{z}_k = h_k(X(t_k)).$$

The following definitions are made

$$F(t) = \frac{\partial f}{\partial X} \bigg|_{\underline{X} = \hat{X}(t_1)}$$

$$H(t) = \frac{\partial h}{\partial X} \bigg|_{\underline{X} = \hat{X}(t_1)}.$$

The following nonlinear differential equations are integrated by Runge-Kutta Gill fourth order numerical integration over the interval (t_0, t_1) .

State Estimate

$$\dot{\underline{X}} = f(\underline{X}(t), t) \quad \underline{X}(t_0) = \underline{X}_0$$

where $f(\underline{X}(t), t)$ is the dynamic model

State Transition Matrix

$$\frac{\partial \phi(t, t_0)}{\partial t} = F(t)\phi(t, t_0), \quad \phi(t_0, t_0) = I$$

Process Noise Covariance

$$\frac{\partial Q(t)}{\partial t} = F(t)Q(t) + Q(t)F(t) + Q_0, \quad Q(t_0) = 0.$$

Once the state transition matrix differential equation is integrated over the interval (t_0, t_1) , the state covariance may be propagated forward to t_1 by the equation

The measurement residual is defined as

$$P(t_1) = \phi(t_1, t_0)P(t_0)\phi^T(t_1, t_0) + Q(t_1).$$

where

$\underline{z}(t_1)$ = the respective sensor measurement model output at time t_1

$$\tilde{z}(t_1) = \underline{z}(t_1) - \hat{z}(t_1) = \underline{z}(t_1) - h(\hat{X}(t_1))$$

Under normal Kalman filter mechanizations, the optimal gain, state estimate update, and state estimate covariance update equations at the measurement time are respectively

$$\begin{aligned}k(t_1) &= P(t_1)H(t_1)(H(t_1)P(t_1)H^T(t_1) + R(t_1))^{-1} \\ \hat{\underline{X}}^+(t_1) &= \hat{\underline{X}}(t_1) + k(t_1)\hat{\underline{Z}}(t_1) \\ P^+(t_1) &= (I - k(t_1)H(t_1))P(t_1)\end{aligned}$$

However, with the Carlson square root mechanization the square root of the state estimate covariance $P(t_1)$ is found by Cholesky decomposition so

$$S(t_1)S(t_1)^T = P(t_1)$$

$S(t_1)$ is then used along with $H(t_1)$ to find the optimal gain and state estimate update. The state estimated covariance update is found in the square root form $S^+(t_1)$ and the reconstruction is implemented by

$$P^+(t_1) = S^+(t_1)S^+(t_1)^T.$$

This process required the restrictions that measurement vectors be decomposed to scalar form and the measurement noise covariance be diagonal (uncorrelated elements). The latter appears to be true for our measurement schemes and coordinate frames used. The former is mechanized by using the elements of the measurement vector $\underline{Z}(t_1)$ in sequence without advancing time from t_1 . This requires the use of the elements of $H(t_1)$ by rows.

The primary function of the remainder of this chapter is to develop the expressions for $H(t_1)$ for each measurement sensor that may be used in the filter.

a. General Formulation

The estimation state vector is

$$\underline{X} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, q_0, q_1, q_2, q_3]^T.$$

The measurement model for each instrument has been previously described and has the form

$$\underline{Z}_j = h_j(\underline{X}) + \underline{V}_j, \quad j = G, S, L$$

where

G = GPS measurement,
S = Star tracker measurement,
L = Landmark tracker measurement.

The estimated measurement will therefore have the form

$$\underline{Z}_j = h_j(\underline{X})$$

where

\underline{Z} and \underline{X} are estimates of the respective vector variables.

Therefore

$$H_j = \frac{\partial \hat{Z}_j}{\partial \underline{X}}$$

where

\underline{Z}_j as a function of \underline{X} has been developed as previously discussed.

b. GPS Measurement Distribution Matrix - H_G

$$\underline{Z}_G = h_G(\underline{X}) + \underline{V}_G$$

so

$$\hat{\underline{Z}}_G = h_G(\hat{\underline{X}}) = \begin{bmatrix} I & 0 & 0' \\ 0 & I & 0' \end{bmatrix} \hat{\underline{X}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}.$$

Therefore,

$$H_G = \frac{\partial \hat{\underline{Z}}}{\partial \hat{\underline{X}}} = \begin{bmatrix} I & 0 & 0' \\ 0 & I & 0' \end{bmatrix}, \text{ where}$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad 0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad 0' = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

c. Star Tracker Measurement Distribution Matrix - H_S

$$\underline{Z}_S = h_S(\underline{X}) + \underline{V}$$

and

$$\hat{\underline{Z}}_S = h_S(\hat{\underline{X}}) = \begin{bmatrix} \Delta H_S \\ \Delta V_S \end{bmatrix} = \begin{bmatrix} \sin^{-1}(\hat{U}_{Sy} / \cos(\sin^{-1} \hat{U}_{Sx})) \\ \sin^{-1} \hat{U}_{Sx} \end{bmatrix} = \begin{bmatrix} \hat{U}_{Sy} \\ \hat{U}_{Sx} \end{bmatrix}$$

where U_{sx} and U_{sy} are elements of an estimated unit vector to a sighted star, \underline{U}_s , in the coordinates of the tracker of interest. (For small angles $\Delta H_S = \hat{U}_{Sy}$ and $\Delta V_S = U_{Sx}$)

To simulate the measurement process as closely as possible, a unit vector to a star is first established in inertial space. This would be the normal star measurement. This unit vector is established using exact star tracker alignment and exact body orientation and is then transformed into tracker coordinates using estimated parameters to develop the estimated measurement

$$\underline{U}_s = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \text{Ideal unit vector along the S/T boresight in S/T coordinates}$$

Transformation to inertial space

$$\underline{U}_T = I^T_S \underline{U}_s = I^T_B B^T_S \underline{U}_s$$

where

I^T_B = exact body to inertial transformation

$$= f_1(q_0, q_1, q_2, q_3) \quad (q_i = \text{body quaternion elements})$$

B^T_S = exact S/T to body transformation

$$= f_2(\alpha, \beta, \gamma, \{0\}) \quad (\alpha, \beta, \gamma = \text{misalignment elements} \\ \{0\} = \text{nominal orientation array}).$$

\underline{U}_I is the simulated unit vector to the star using exact attitude information. Now transforming this vector to star tracker coordinates using the estimated attitude information yields:

$$\underline{U}_s = S^T_I \underline{U}_I = S^T_B \hat{B}^T_I \underline{U}_I = S^T_B \hat{B}^T_I I^T_B B^T_S \underline{U}_s$$

where

$$\hat{B}^T_I = f_3(\hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3), \quad (\hat{q}_i = \text{estimates of } q_i),$$

$$S^T_B = f_4(\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \{0\}), \quad (\hat{\alpha}, \hat{\beta}, \hat{\gamma} = \text{estimates of } \alpha, \beta, \gamma)$$

where

$\hat{\alpha}$, $\hat{\beta}$, and $\hat{\gamma}$ are a priori estimates of misalignment angles of the star tracker of interest relative to the nominal alignment. These parameters may be estimated at a later data but will be used here as implicit consider parameters.

$\{0\}$ is a nominal orientation array.

Since \underline{z}_s is a nonlinear function of the elements of the state estimate \underline{x} , H_s must be found by

$$H_s = \frac{\partial \hat{z}_s}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \Delta H_s}{\partial \underline{x}} \\ \frac{\partial \Delta V_s}{\partial \underline{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \underline{x}} \sin^{-1}(\hat{u}_{sy}/\cos(\sin^{-1}\hat{u}_{sx})) \\ \frac{\partial}{\partial \underline{x}} \sin^{-1}\hat{u}_{sx} \end{bmatrix}$$

$$\approx \begin{bmatrix} \frac{\partial \hat{u}_{sy}}{\partial \underline{x}} \\ \frac{\partial \hat{u}_{sx}}{\partial \underline{x}} \end{bmatrix}$$

Where

$$\frac{\partial \Delta V_s}{\partial \hat{x}_1} = \frac{1}{\sqrt{1-(\hat{u}_{sx})^2}} \frac{\partial \hat{u}_{sx}}{\partial \hat{x}_1} \approx \frac{\partial \hat{u}_{sx}}{\partial \hat{v}_1}$$

$$\frac{\partial \Delta H_s}{\partial \hat{x}_1} = \frac{1}{\sqrt{1-\left(\frac{\hat{u}_{sy}}{\cos(\sin^{-1}\hat{u}_{sx})}\right)^2}} \frac{\cos(\sin^{-1}\hat{u}_{sx}) \frac{\partial \hat{u}_{sy}}{\partial \hat{x}_1} + \hat{u}_{sy} \cdot \hat{u}_{sx} \cdot \frac{1}{\sqrt{1-(\hat{u}_{sx})^2}} \frac{\partial \hat{u}_{sx}}{\partial \hat{x}_1}}{(\cos(\sin^{-1}\hat{u}_{sx}))^2}$$

$$\approx \frac{\partial \hat{u}_{sy}}{\partial \hat{x}_1}$$

$$\hat{x}_1 = \hat{x}, \hat{y}, \hat{z}, \hat{\dot{x}}, \hat{\dot{y}}, \hat{\dot{z}}, \hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3.$$

Since $I^T S$ is independent of any elements of the estimated state, consider

$$I^T S = I^T B B^T S = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = [a_{ij}].$$

Therefore

$$\underline{U}_I = \underline{I}^T \underline{S} \underline{U}_s = \begin{bmatrix} a_{1j} \\ a_{2j} \\ a_{3j} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}.$$

So

$$\underline{\hat{U}}_s = \underline{S}^T \underline{B} \underline{\hat{B}}^T \underline{\hat{I}}_1 \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} = \underline{S}^T \underline{B} \begin{bmatrix} \hat{q}_{11} & \hat{q}_{12} & \hat{q}_{13} \\ \hat{q}_{21} & \hat{q}_{22} & \hat{q}_{23} \\ \hat{q}_{31} & \hat{q}_{32} & \hat{q}_{33} \end{bmatrix} \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

where

$$\hat{q}_{11} = \hat{q}_0^2 + \hat{q}_1^2 - \hat{q}_2^2 - \hat{q}_3^2$$

$$\hat{q}_{12} = 2(\hat{q}_1 \hat{q}_2 + \hat{q}_3 \hat{q}_0)$$

$$\hat{q}_{13} = 2(\hat{q}_1 \hat{q}_3 - \hat{q}_2 \hat{q}_0)$$

$$\hat{q}_{21} = 2(\hat{q}_1 \hat{q}_2 - \hat{q}_3 \hat{q}_0)$$

$$\hat{q}_{22} = \hat{q}_0^2 - \hat{q}_1^2 + \hat{q}_2^2 - \hat{q}_3^2$$

$$\hat{q}_{23} = 2(\hat{q}_2 \hat{q}_3 + \hat{q}_1 \hat{q}_0)$$

$$\hat{q}_{31} = 2(\hat{q}_1 \hat{q}_3 + \hat{q}_2 \hat{q}_0)$$

$$\hat{q}_{32} = 2(\hat{q}_2 \hat{q}_3 - \hat{q}_1 \hat{q}_0)$$

$$\hat{q}_{33} = \hat{q}_0^2 - \hat{q}_1^2 - \hat{q}_2^2 + \hat{q}_3^2.$$

Since α , β , and γ are not being estimated in the filter, they are considered constant and imbedded in $\underline{S}^T \underline{B}$ as

$$\underline{S}^T \underline{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} b_{1j} \\ b_{2j} \\ b_{3j} \end{bmatrix}.$$

Therefore

$$\hat{U}_s = \begin{bmatrix} b_{1j} \\ b_{2j} \\ b_{3j} \end{bmatrix} \begin{bmatrix} \hat{q}_{1j} \\ \hat{q}_{2j} \\ \hat{q}_{3j} \end{bmatrix} = \begin{bmatrix} \hat{U}_{sx} \\ \hat{U}_{sy} \\ \hat{U}_{sz} \end{bmatrix}$$

Since U_{sz} is not observed by the star tracker, it is dropped from the measurement vector.

$$\begin{aligned} \hat{U}_{sx} = & b_{11} (\hat{q}_{11}a_{13} + \hat{q}_{12}a_{23} + \hat{q}_{13}a_{33}) \\ & + b_{12} (\hat{q}_{21}a_{13} + \hat{q}_{22}a_{23} + \hat{q}_{23}a_{33}) \\ & + b_{13} (\hat{q}_{31}a_{13} + \hat{q}_{32}a_{23} + \hat{q}_{33}a_{33}) \end{aligned}$$

$$\begin{aligned} \hat{U}_{sy} = & b_{21} (\hat{q}_{11}a_{13} + \hat{q}_{12}a_{23} + \hat{q}_{13}a_{33}) \\ & + b_{22} (\hat{q}_{21}a_{13} + \hat{q}_{22}a_{23} + \hat{q}_{23}a_{33}) \\ & + b_{23} (\hat{q}_{31}a_{13} + \hat{q}_{32}a_{23} + \hat{q}_{33}a_{33}) \end{aligned}$$

Define

$$\hat{U}_s = \begin{bmatrix} \hat{U}_{sx} \\ \hat{U}_{sy} \end{bmatrix}$$

$$\frac{\partial \hat{U}_{sx}}{\partial \hat{x}_i} = \frac{\partial \hat{U}_{sy}}{\partial \hat{x}_i} = 0 \quad \text{for } \hat{x}_i = \hat{x}, \hat{y}, \hat{z}, \hat{\dot{x}}, \hat{\dot{y}}, \hat{\dot{z}}$$

$$\begin{aligned} \frac{\partial \hat{U}_{sx}}{\partial \hat{q}_1} = & b_{11} (a_{13} \frac{\partial \hat{q}_{11}}{\partial \hat{q}_1} + a_{23} \frac{\partial \hat{q}_{12}}{\partial \hat{q}_1} + a_{33} \frac{\partial \hat{q}_{13}}{\partial \hat{q}_1}) \\ & + b_{12} (a_{13} \frac{\partial \hat{q}_{21}}{\partial \hat{q}_1} + a_{23} \frac{\partial \hat{q}_{22}}{\partial \hat{q}_1} + a_{33} \frac{\partial \hat{q}_{23}}{\partial \hat{q}_1}) \\ & + b_{13} (a_{13} \frac{\partial \hat{q}_{31}}{\partial \hat{q}_1} + a_{23} \frac{\partial \hat{q}_{32}}{\partial \hat{q}_1} + a_{33} \frac{\partial \hat{q}_{33}}{\partial \hat{q}_1}) \end{aligned}$$

$$\begin{aligned} \frac{\partial \hat{U}_{sy}}{\partial \hat{q}_1} = & b_{21} (a_{13} \frac{\partial \hat{q}_{11}}{\partial \hat{q}_1} + a_{23} \frac{\partial \hat{q}_{12}}{\partial \hat{q}_1} + a_{33} \frac{\partial \hat{q}_{13}}{\partial \hat{q}_1}) \\ & + b_{22} (a_{13} \frac{\partial \hat{q}_{21}}{\partial \hat{q}_1} + a_{23} \frac{\partial \hat{q}_{22}}{\partial \hat{q}_1} + a_{33} \frac{\partial \hat{q}_{23}}{\partial \hat{q}_1}) \\ & + b_{23} (a_{13} \frac{\partial \hat{q}_{31}}{\partial \hat{q}_1} + a_{23} \frac{\partial \hat{q}_{32}}{\partial \hat{q}_1} + a_{33} \frac{\partial \hat{q}_{33}}{\partial \hat{q}_1}) \end{aligned}$$

where

$$\frac{\partial \hat{q}_{11}}{\partial \hat{q}_0} = 2\hat{q}_0 \quad \frac{\partial \hat{q}_{12}}{\partial \hat{q}_0} = 2\hat{q}_3 \quad \frac{\partial \hat{q}_{13}}{\partial \hat{q}_0} = -2\hat{q}_2$$

$$\frac{\partial \hat{q}_{21}}{\partial \hat{q}_0} = -2\hat{q}_3 \quad \frac{\partial \hat{q}_{22}}{\partial \hat{q}_0} = 2\hat{q}_0 \quad \frac{\partial \hat{q}_{23}}{\partial \hat{q}_0} = 2\hat{q}_1$$

$$\frac{\partial \hat{q}_{31}}{\partial \hat{q}_0} = 2\hat{q}_2 \quad \frac{\partial \hat{q}_{32}}{\partial \hat{q}_0} = -2\hat{q}_1 \quad \frac{\partial \hat{q}_{33}}{\partial \hat{q}_0} = 2\hat{q}_0$$

Or in alternate form

$$\frac{\partial}{\partial \hat{q}_0} [\hat{q}_{ij}] = 2 \begin{bmatrix} \hat{q}_0 & \hat{q}_3 & -\hat{q}_2 \\ -\hat{q}_3 & \hat{q}_0 & \hat{q}_1 \\ \hat{q}_2 & -\hat{q}_1 & \hat{q}_0 \end{bmatrix}.$$

Similarly

$$\frac{\partial}{\partial \hat{q}_1} [\hat{q}_{ij}] = 2 \begin{bmatrix} \hat{q}_1 & \hat{q}_2 & -\hat{q}_3 \\ \hat{q}_2 & -\hat{q}_1 & \hat{q}_0 \\ \hat{q}_3 & -\hat{q}_0 & -\hat{q}_1 \end{bmatrix}.$$

$$\frac{\partial}{\partial \hat{q}_2} [\hat{q}_{ij}] = 2 \begin{bmatrix} -\hat{q}_2 & \hat{q}_1 & -\hat{q}_0 \\ \hat{q}_1 & \hat{q}_2 & \hat{q}_3 \\ \hat{q}_0 & \hat{q}_3 & -\hat{q}_2 \end{bmatrix}$$

$$\frac{\partial}{\partial \hat{q}_3} [\hat{q}_{ij}] = 2 \begin{bmatrix} -\hat{q}_3 & \hat{q}_0 & \hat{q}_1 \\ -\hat{q}_0 & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_1 & \hat{q}_2 & \hat{q}_3 \end{bmatrix}.$$

So

$$\frac{\partial \hat{U}_s}{\partial \hat{x}_1} = \begin{bmatrix} 0 & 0 & \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} \frac{\partial}{\partial \hat{q}_1} \begin{bmatrix} \hat{q}_{1j} \end{bmatrix} \begin{bmatrix} a_{1j} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

$$\hat{U}_s = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} \begin{bmatrix} \hat{q}_{1j} \end{bmatrix} \begin{bmatrix} a_{1j} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Finally

$$H_s = \frac{\partial \hat{z}}{\partial \hat{x}} = \begin{bmatrix} \frac{\partial \Delta H}{\partial x} & \frac{\partial \Delta H}{\partial y} & \frac{\partial \Delta H}{\partial z} & \frac{\partial \Delta H}{\partial \hat{x}} & \frac{\partial \Delta H}{\partial \hat{y}} & \frac{\partial \Delta H}{\partial \hat{z}} & \frac{\partial \Delta H}{\partial \hat{q}_0} & \frac{\partial \Delta H}{\partial \hat{q}_1} & \frac{\partial \Delta H}{\partial \hat{q}_2} & \frac{\partial \Delta H}{\partial \hat{q}_3} \\ \frac{\partial \Delta V}{\partial x} & \frac{\partial \Delta V}{\partial y} & \frac{\partial \Delta V}{\partial z} & \frac{\partial \Delta V}{\partial \hat{x}} & \frac{\partial \Delta V}{\partial \hat{y}} & \frac{\partial \Delta V}{\partial \hat{z}} & \frac{\partial \Delta V}{\partial \hat{q}_0} & \frac{\partial \Delta V}{\partial \hat{q}_1} & \frac{\partial \Delta V}{\partial \hat{q}_2} & \frac{\partial \Delta V}{\partial \hat{q}_3} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & \frac{\partial \Delta H}{\partial q_0} & \frac{\partial \Delta H}{\partial q_1} & \frac{\partial \Delta H}{\partial q_2} & \frac{\partial \Delta H}{\partial q_3} \\ \frac{\partial \Delta V}{\partial q_0} & \frac{\partial \Delta V}{\partial q_1} & \frac{\partial \Delta V}{\partial q_2} & \frac{\partial \Delta V}{\partial q_3} \end{bmatrix}.$$

The approximation equation will be used initially and expanded later.

d. Landmark Tracker Distribution Matrix - H_L

$$\underline{Z}_L = h_L(\underline{X}) + \underline{N}_L$$

and

$$\hat{\underline{Z}}_L = h_L(\hat{\underline{X}}) = \begin{bmatrix} \Delta H_L \\ \Delta V_L \end{bmatrix} = \begin{bmatrix} \sin^{-1}(\hat{U}_L y / \cos(\sin^{-1} \hat{U}_L x)) \\ \sin^{-1} \hat{U}_L x \end{bmatrix}$$

$$= \begin{bmatrix} \hat{U}_L y \\ \hat{U}_L x \end{bmatrix}$$

where $\hat{U}_L x$ and $\hat{U}_L y$ are elements of an estimated unit vector to the sighted landmark, U_L , in landmark tracker coordinates. (Again for small angles $\Delta H_L \approx \hat{U}_L x$ and $\Delta V_L \approx \hat{U}_L y$.)

The simulated measurement is made by unitizing the vector from the spacecraft to the Landmark. The result is then transformed to tracker coordinates through the estimate of the transformation from inertial space to landmark track coordinates, ${}_{\ell}T_I$.

The estimated measurement vector in inertial space is formed by

$$\hat{\underline{M}}_I = ({}_IT_E \underline{L}_E) - \hat{\underline{P}}_I$$

where

\underline{L}_E = the landmark position vector in earth fixed coordinates;

$\hat{\underline{P}}_I$ = the spacecraft position vector estimate in inertial space.

Transforming this to the best knowledge of landmark tracker coordinates yields:

$$\begin{aligned}\hat{\underline{M}}_{\ell} &= {}_{\ell}\hat{T}_I (({}_IT_E \underline{L}_E) - \hat{\underline{P}}_I) \\ &= {}_{\ell}\hat{T}_B B\hat{T}_I (({}_IT_E \underline{L}_E) - \hat{\underline{P}}_I)\end{aligned}$$

where

$$B\hat{T}_I = f_3 (\hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3)$$

$${}_{\ell}\hat{T}_B = f_5 (\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \{0\}), (\hat{\alpha}, \hat{\beta}, \hat{\gamma} = \text{estimates of the tracker misalignment angles. } \{0\} = \text{nominal orientation array}).$$

Consider that

$${}_IT_E \underline{L}_E = \underline{L}_I = \underline{L}$$

$$\hat{\underline{P}}_I = \hat{\underline{P}} = f_6 (\hat{x}, \hat{y}, \hat{z})$$

$$B\hat{T}_I = [\hat{q}_{ij}] \text{ (defined in star tracker section)}$$

$${}_{\ell}\hat{T}_B = [\hat{a}_{ij}].$$

So

$$\hat{\underline{M}}_{\ell} = [\hat{a}_{ij}] [\hat{q}_{ij}] (\underline{L} - \hat{\underline{P}}).$$

Unitizing $\hat{\underline{M}}_{\ell}$ into a unit vector

$$\hat{\underline{U}}_{\ell} = \frac{\hat{\underline{M}}_{\ell}}{|\hat{\underline{M}}_{\ell}|} = \begin{bmatrix} \hat{U}_{\ell x} \\ \hat{U}_{\ell y} \\ \hat{U}_{\ell z} \end{bmatrix} \text{ where } |\hat{\underline{M}}_{\ell}| = (\hat{\underline{M}}_{\ell}^T \hat{\underline{M}}_{\ell})^{1/2}.$$

Since \underline{Z} is very similar to \underline{Z}_s , H may be found directly as

$$H_{\ell} = \frac{\partial \hat{Z}_{\ell}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial \Delta H_{\ell}}{\partial \underline{x}} \\ \frac{\partial \Delta V_{\ell}}{\partial \underline{x}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial}{\partial \underline{x}} \sin^{-1}(\hat{U}_{\ell y} / \cos(\sin^{-1} \hat{U}_{\ell x})) \\ \frac{\partial}{\partial \underline{x}} \sin^{-1} \hat{U}_{\ell x} \end{bmatrix} \approx \begin{bmatrix} \frac{\partial \hat{U}_{\ell y}}{\partial \underline{x}} \\ \frac{\partial \hat{U}_{\ell x}}{\partial \underline{x}} \end{bmatrix}$$

where

$$\frac{\partial \Delta V_{\ell}}{\partial \hat{x}_1} = \frac{1}{\sqrt{1 - (\hat{U}_{\ell x})^2}} \frac{\partial \hat{U}_{\ell x}}{\partial \hat{x}_1} \approx \frac{\partial \hat{U}_{\ell x}}{\partial \hat{x}_1}$$

and

$$\frac{\partial \Delta H_{\ell}}{\partial \hat{x}_1} = \frac{1}{\left(1 - \left(\frac{\hat{U}_{\ell y}}{\cos(\sin^{-1} \hat{U}_{\ell x})}\right)^2\right)^{1/2}} \frac{\cos(\sin^{-1} \hat{U}_{\ell x}) \frac{\partial}{\partial \hat{x}_1} \hat{U}_{\ell y} + \hat{U}_{\ell y} \cdot \hat{U}_{\ell x} \cdot \frac{1}{\sqrt{1 - (\hat{U}_{\ell x})^2}} \cdot \frac{\partial \hat{U}_{\ell x}}{\partial \hat{x}_1}}{(\cos(\sin^{-1} \hat{U}_{\ell y}))^2}$$

$$\approx \frac{\partial \hat{U}_{\ell y}}{\partial \hat{x}_1}$$

$$x_1 = \hat{x}, \hat{y}, \hat{z}, \hat{x}, \hat{y}, \hat{z}, \hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3$$

$$x_i = x, y, z, x, y, z, q_0, q_1, q_2, q_3.$$

Since

$$\hat{U}_{\ell} = \frac{\hat{M}_{\ell}}{|\hat{M}_{\ell}|}$$

$$\frac{\partial \hat{U}_{\ell}}{\partial \hat{x}_1} = \frac{|\hat{M}_{\ell}| \frac{\partial \hat{M}_{\ell}}{\partial \hat{x}_1} - \hat{M}_{\ell} \frac{\partial |\hat{M}_{\ell}|}{\partial \hat{x}_1}}{|\hat{M}_{\ell}|^2} : \text{ a three vector for each } x_i$$

making a total of (3x22)

where

$$\frac{\partial \hat{M}_l}{\partial \hat{x}_1} = \{\hat{a}_{1j}\} \left[\frac{\partial}{\partial \hat{x}_1} \{\hat{q}_{1j}\} (\underline{L} - \hat{P}) + \{\hat{q}_{1j}\} \frac{\partial}{\partial \hat{x}_1} (\underline{L} - \hat{P}) \right] \quad i = 1, 10$$

$$\frac{\partial |\hat{M}_l|}{\partial \hat{x}_1} = \frac{\partial}{\partial \hat{x}_1} (\hat{M}_l^T \hat{M}_l)^{1/2} = \frac{1}{2} (\hat{M}_l^T \hat{M}_l)^{-1/2} \left[\left(\frac{\partial}{\partial \hat{x}_1} \hat{M}_l^T \right) \hat{M}_l + \hat{M}_l^T \frac{\partial}{\partial \hat{x}_1} \hat{M}_l \right] = (\hat{M}_l^T \frac{\partial}{\partial \hat{x}_1} \hat{M}_l) |\hat{M}_l|$$

$$\frac{\partial}{\partial \hat{x}_1} \hat{M}_l^T = \left(\frac{\partial}{\partial \hat{x}_1} \hat{M}_l \right)^T$$

$$\frac{\partial}{\partial \hat{x}_1} \{\hat{q}_{1j}\} = \{0\} \quad \text{for } \hat{x}_1 = x, y, z, \hat{x}, \hat{y}, \hat{z}$$

$$= 2 \begin{bmatrix} \hat{q}_0 & \hat{q}_3 & -\hat{q}_2 \\ -\hat{q}_3 & \hat{q}_0 & \hat{q}_1 \\ \hat{q}_2 & -\hat{q}_1 & \hat{q}_0 \end{bmatrix} \quad \text{for } \hat{x}_1 = q_0$$

$$= 2 \begin{bmatrix} \hat{q}_1 & \hat{q}_2 & +\hat{q}_3 \\ \hat{q}_2 & -\hat{q}_1 & \hat{q}_0 \\ \hat{q}_3 & -\hat{q}_0 & -\hat{q}_1 \end{bmatrix} \quad \text{for } \hat{x}_1 = q_1$$

$$= 2 \begin{bmatrix} -\hat{q}_2 & \hat{q}_1 & -\hat{q}_0 \\ \hat{q}_1 & \hat{q}_2 & \hat{q}_3 \\ \hat{q}_0 & \hat{q}_3 & -\hat{q}_2 \end{bmatrix} \quad \text{for } \hat{x}_1 = q_2$$

$$= 2 \begin{bmatrix} -\hat{q}_3 & \hat{q}_0 & \hat{q}_1 \\ -\hat{q}_0 & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_1 & \hat{q}_2 & \hat{q}_3 \end{bmatrix} \quad \text{for } \hat{x}_1 = q_3$$

and

$$\frac{\partial}{\partial \hat{x}_1} [\underline{L} - \hat{P}] = \underline{0} \quad \text{for } \hat{x}_1 = \hat{x}, \hat{y}, \hat{z}, \hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3$$

$$= \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad \text{for } \hat{x}_1 = \hat{x}$$

$$= \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad \text{for } \hat{x}_1 = \hat{y}$$

$$= \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \text{for } \hat{x}_1 = \hat{z}$$

B. EXTRACTED GCP MEASUREMENTS

The second mode of operation allowed by GCPSIM provides for a mixture of modeled and actual measurements. The GPS and star tracker measurements are modeled as discussed previously. The GCP measurements, however, are extracted from actual imagery obtained from Landsat. For this process, two passes of corrected Landsat imagery over the same area are required. The first pass is used to extract GCPs to be incorporated into the GCP data base discussed in Chapter II. Since the extracted GCPs are used for references, it is desirable that the first pass of imagery be of good quality with as little cloud coverage as possible. It is also necessary that the latitude and longitude of the four corners of the image be known so the latitude and longitude of the individual landmarks can be calculated. Given these constraints, it is possible to construct a GCP data base to be used in the simulation. A program, entitled LMEX, has been developed to aid this process.

The second mode of operation functions much the same way as the first mode. The sequencer defines the type and time of the star tracker and GPS measurements. The GCP data base containing the number of scan lines to the next GCP is consulted to see if the GCP will be encountered prior to the measurement; if there is not enough time, the measurement defaults to a GCP sighting. The primary difference between the two modes of operation is that prior to exiting from the sequencer, an interrupt is sent via a parallel data bus to a separate task hosted on the PDP 11/70. The interrupt contains the number of lines to be scanned prior to measurement time and a flag identifying whether a GCP sighting will be available or not.

The entire purpose of the second program is to perform all the functions of the registration processor and to simulate the sensor front end.

The program has been separated into three tasks running independently on the PDP 11/70 at different priorities. Intertask communication is performed through a shareable global area, and synchronization is controlled using global event flags. The three tasks are described as:

- 1) SCROLL - Accepts an interrupt from the VAX 11/780 with the number of lines to be scrolled and a flag indicating whether a GCP sighting is available. SCROLL also performs the actual scrolling of imagery across a video monitor to simulate the acquisition of imagery;
- 2) SCOUNT - When enabled by SCROLL between each scan line, SCOUNT updates the line counter, which indicates the distance to the search area. If the line counter is 0 or negative, SCOUNT also stores the search area imagery as the data are scanned;

- 3) CORR - SCOUNT initializes CORR when the entire search area has been read into memory. CORR then performs the SSDA registration process using the reference GCP extracted from the first pass of imagery. CORR also passes the registration vector back to SCROLL, which passes it to GCPSIM.

The interrelationships of the tasks are illustrated in Figure III-6 and the process is illustrated in Figures III-7. For a detailed description of the software, consult Volume III.

Several problems were encountered when developing this simulation. First, both passes of imagery must be corrected for distortions because the distortions associated with the multispectral scanner (MSS) are too dynamic to be modeled. This problem will not be encountered with either the thematic mapper or the MLA because distortions are deterministic. If uncorrected imagery is used, the location of the extracted GCPs cannot be accurately determined, and the registration vector will have a large component caused by this uncertainty and the distortion of the second pass. As a result, the filter has poor convergence characteristics because the effective noise, rather than being 0.1 pixel, is now on the order of 10 pixels. The imagery acquired as GFE was uncorrected imagery and therefore could not be used.

The second problem encountered is that the true vehicle state at the time of scene exposure is required. However, if the imagery has been corrected, this state is no longer valid because the errors associated with vehicle state have been extracted.

In an operational system, these problems will not be encountered because the distortions associated with the MLA are totally deterministic so the location of the GCPs are well-known functions of spacecraft state.

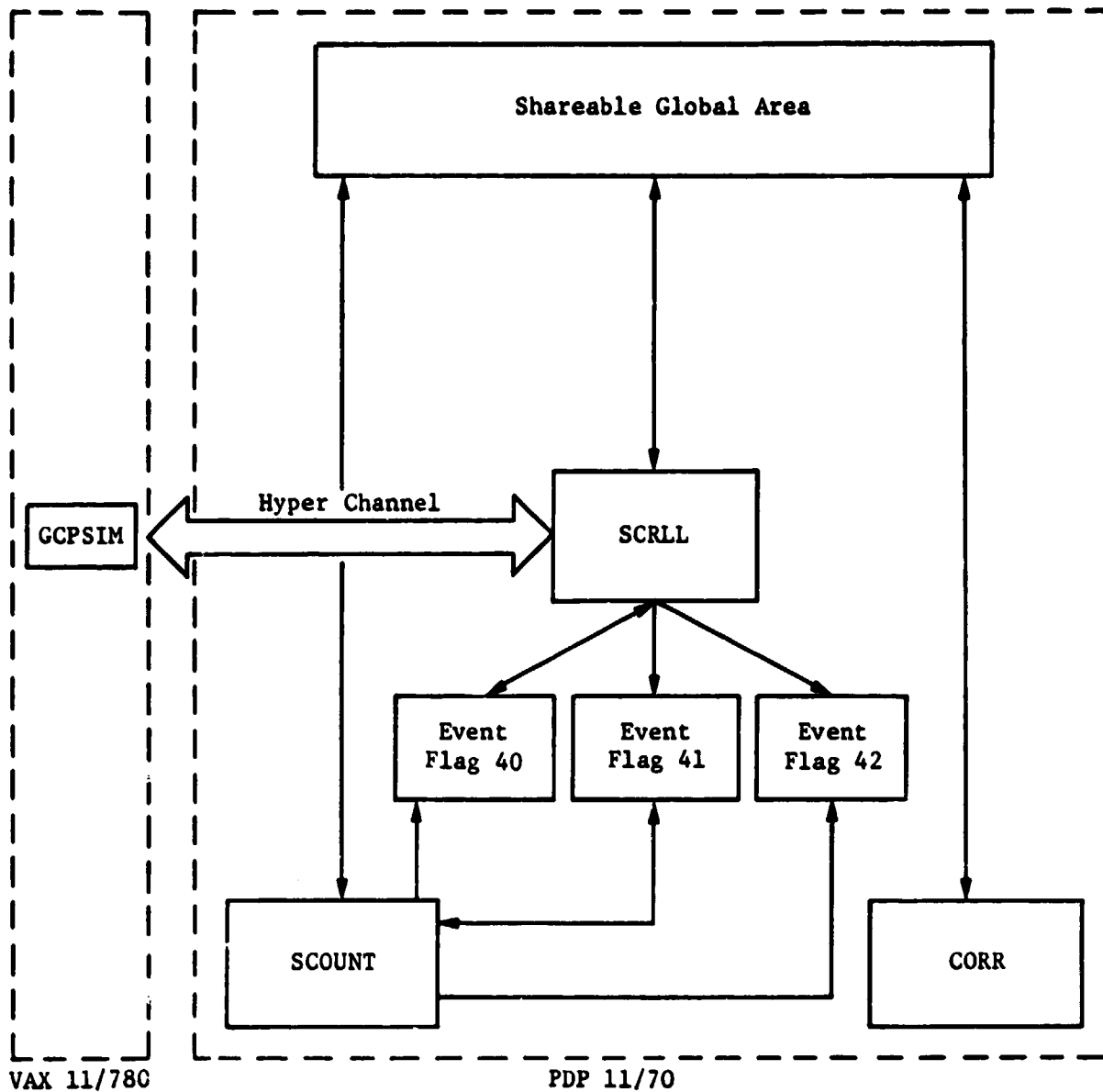


Figure III-6 Correlation - Scroll Program Structure

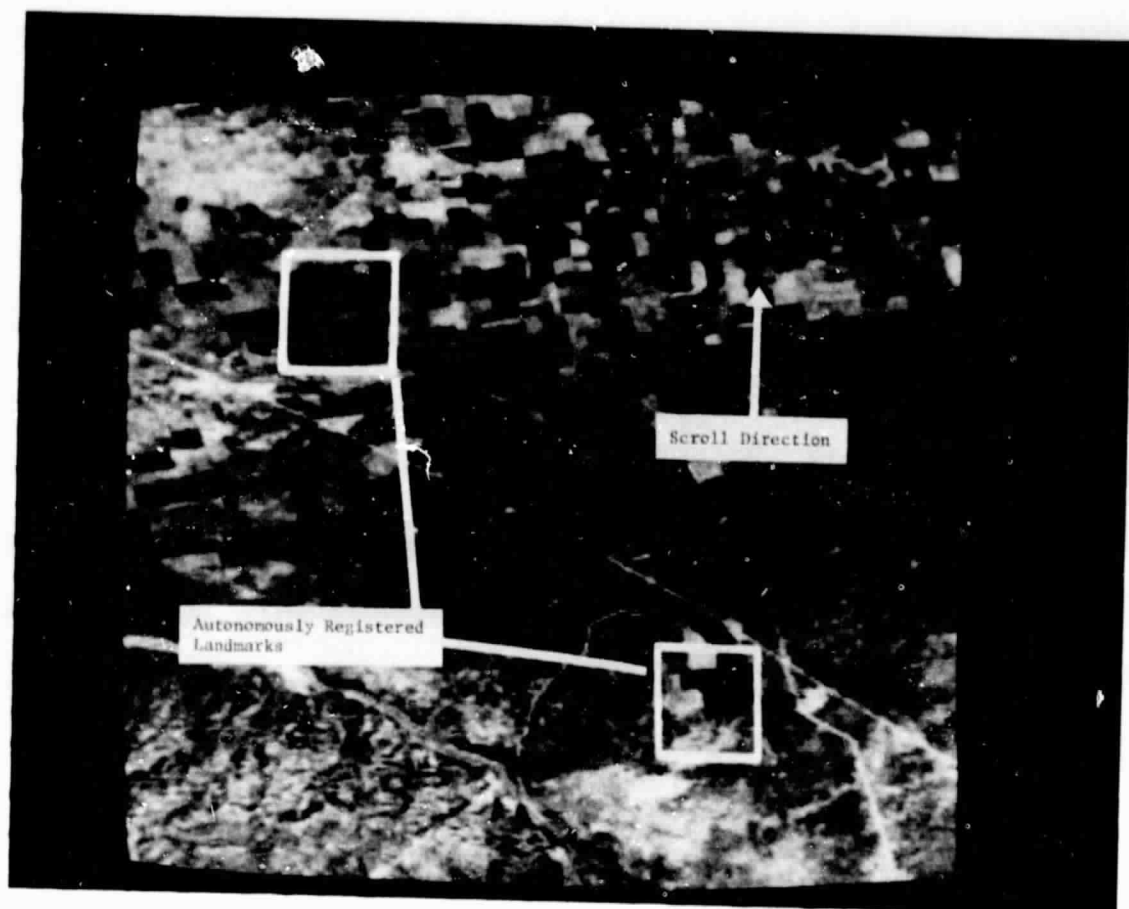


Figure III-7 Automatic Landmark Extraction Simulation

ORIGINAL PAGE IS
OF POOR QUALITY

IV. GCP ANALYSIS

The GCP simulation system configuration discussion in chapter II will not be repeated here. Other areas such as assumed orbital characteristics, vehicle motion, sensor error budget and the results of parametric studies will be discussed in detail.

For the purpose of this study, the spacecraft orbit was assumed to be circular with an inclination to the equatorial plane at 80 degrees. For ease of setup and interpretation of results, the beginning of each simulation run was taken to be at a positive ascending node with the inertial coordinate frame coincident with the earth's coordinate frame. This did not detract from the generality of the results since specific scenes on the surface or specific star catalogs were not used in the simulation.

The vehicle's motion was initially intended to be that of table lookup of vehicle body rates about each vehicle axis, followed by integration of these rates to determine attitude. This, however, was changed to a forced nadir-oriented vehicle through derivation of a vehicle body rate vector found by computing the cross product of the true vehicle velocity and position vectors. This rate vector was used as input to both the gyro model and the true attitude propagator where true attitude was obtained through integration of the body rates.

A. ERROR BUDGET

1. Attitude Reference Unit

The attitude reference unit for this simulation was assumed to be of the NASA standard type (DRIDU II). It consists of three nominally orthogonal gyros whose characteristics are fairly well documented. The specifications for these gyros were studied and the terms applicable to the assumed system configuration and environment were used. The parameters specified and used are outlined in Table IV-1.

2. Star Trackers

Each of the two star trackers assumed to be on board the vehicle are of the NASA standard type. For the purposes of this simulation, the optical axes of the two devices were assumed to be orthogonally oriented in a plane containing the vehicle position vector and perpendicular to the vehicle velocity vector. Documentation, including the OADS and existing specifications, was studied and the error terms felt to be applicable to this system configuration and environment were used. These parameters are outlined in Table IV-2.

Table IV-1 Attitude Reference Unit Error Coefficients

Parameter	Value of Conventional Units, 1σ	Value Used, 1σ	Source
Gyro Long-Term Bias Uncertainty	0.017 deg/h	0.08×10^{-6} rad/s	Spec
Scale Factor Uncertainty	90 ppm	90.0×10^{-6}	Spec
Nonorthogonality of Mounting Package Alignment Uncertainty Relative to Mounting Surface	2 arc-s	10×10^{-6} rad	Assumed
Each Axis	10 arc-s	50×10^{-6}	Assumed
Gyro Noise Equivalent Angle	1.3 arc-s	10×10^{-6} rad/s	Spec
Absolute Bias	2 deg/h	10×10^{-6} rad/s	Spec
Absolute Scale Factor	0.5%	0.005	Spec

Table IV-2 Star Tracker Error Coefficients

Parameter	Value of Conventional Units, 1σ	Value Used, 1σ	Source
Bias, Each Axis	10 arc-s	50×10^{-6} rad	Spec
Quantization	1 arc-s	5×10^{-6} rad	OADS
Misalignment	5 arc-s	25×10^{-6} rad	OADS
Each Axis	1 arc-s	5×10^{-6} rad	OADS
Star Catalog	1 arc-s	5×10^{-6} rad	OADS

3. Global Positioning System (GPS)

The GPS is assumed to be available to the spacecraft virtually continuously. The update interval using GPS was predominantly 5 seconds; however, variations have been made in this parameter to study its affect on system performance. The GPS system error parameter assumed for the purposes of this simulation are presented in Table IV-3.

Table IV-3 GPS Error Coefficients

Parameter	Value of Conventional Units, 1σ	Value Used, 1σ	Source
Velocity Bias	0	0	Assumed
Velocity Noise	0.005 m/s	6×10^{-6} km/s	Assumed
Position Bias	0.04 m	4×10^{-5} km	Assumed
Position Noise	4.0 m	4×10^{-3} km	Assumed

4. Landmark Tracker (LMT)

The LMT was assumed to have its optical axis colinear with the nadir pointing axis of the spacecraft. The look angle to the landmark is in the plane containing the optical axis, perpendicular to the vehicle velocity vector. It has a maximum value of 10 degrees with two modes of operation. These modes are user-selectable to take on either fixed specified look angles or uniformly distributed look angles of a specified maximum magnitude. Table IV-4 outlines the error characteristics assumed for the LMT.

Table IV-4 Landmark Tracker Error Coefficients

Parameter	Value Conventional Units 1σ	Value Used 1σ	Source
Bias, Each Axis	2 arc-s	10×10^{-6} rad	Assumed
Quantization Each Axis	1 arc-s	5×10^{-6} rad	Assumed
Misalignment (three axes)	5 arc-s	25×10^{-6} rad	Assumed
Landmark Location Uncertainty	1 arc-s	5×10^{-6} rad	Assumed

B. PARAMETRIC STUDIES

The intent of the parametric studies was to determine the system's sensitivity to such parameters as LMT update interval, look angle mode and range, GPS update interval and accuracy, and recovery time after a long period without LMT updates with and without star tracker backup. The study was progressive in the above order using as parameters the values that appeared to be optimum from the previous study. With the exception of the study of system sensitivity to the GPS accuracy, the error budget referred to was used uniformly for all studies.

Table IV-5 outlines the cases studied by title and most significant parameters. Plots showing the results of these runs have been included at the end of this chapter.

1. Sensitivity to GCP Update Interval

The sensitivity to GCP update interval was investigated by examining the system performance parameters of estimated position, velocity, and attitude accuracy after system convergence for GCP interval variation from 5 to 320 seconds in factors of 2. These are respectively cases 1 through 7. While making these variations, the GPS update interval was held at 5 seconds and the error budget held to the values previously

specified. Both random (+10-deg maximum) and fixed (0-deg) look angles were considered for the landmark tracker pointing for each case presented. No star tracker measurements were used during this study.

Table IV-5 Cases Studied

Title	GCP Update Interval, s	GPS Update Interval, s	Remarks
GCP Update Interval Sensitivity			
CASE 1	5	5	CASE 1 through CASE 7 were run with +10-deg random look angles as well as a 0-degree fixed look angle.
CASE 2	10	5	
CASE 3	20	5	
CASE 4	40	5	
CASE 5	80	5	
CASE 6	160	5	
CASE 7	320	5	
Look Angle Sensitivity			
CASE 9	80	5	0-deg fixed look angle.
CASE 10	80	5	2.5-deg fixed look angle.
CASE 12	80	5	5-deg fixed look angle.
CASE 14	80	5	10-deg fixed look angle.
CASE 17	80	5	+2.5-deg random look angle.
CASE 18	80	5	+5-deg random look angle.
GPS Sensitivity Update Interval			
CASE 21	80	10	CASE 21 through CASE 26 were run with +10-deg look angles.
CASE 22	80	20	
CASE 23	80	40	
CASE 24	80	80	
Accuracy			
CASE 25	80	5	GPS velocity error increased by 4X. GPS position error increased by 4X.
CASE 26	80	5	
Recovery Sensitivity			
CASE 30	80	5	+10-deg random look angles; linear cloud table; default to star trackers.
CASE 31	80	5	+10-deg random look angles; GCP shut down for 1000 s with default to star trackers.
CASE 32	80	5	+10-deg random look angle; GCP shut down for 1000 s without default to star trackers.

The standard deviation of error is found by identifying the peak attitude and position errors after filter convergence. These values are taken as the 3σ attitude error, implying that the error standard deviation σ is the peak error value divided by 3. Table IV-6 lists these standard deviation values.

Table IV-6 Results of GCP Update Interval Sensitivity

GCP Update Interval, s	Peak Position Error m	Peak Attitude Error, arc-s	Bounds of Implied Surface Error, m 1 σ
5	7.0	16	18-21
10	6.5	14	16-18
20	7.0	11	15-17
40	7.0	10	12-14
80	6.5	16	18-20
160	9.0	15	17-20
320	6.0	15	17-19

Initial runs showed that when the GCP look angle was held at nadir, attitude error convergence was poor. After some analysis, this phenomenon was shown to be an observability problem. When uniformly distributed random look angles with peak magnitudes of 10 degrees were implemented, convergence became more rapid and definitive. For this reason the results were taken on the basis of random look angles. The results showing the attitude error convergence with look angles fixed at zero are shown for information only. The fixed look angle cases are indicated on the curves.

In the way of quantitatively evaluating the results, the most desirable parameter to consider is the surface location error δP resulting from the combined navigational position error δP_1 of the spacecraft and the implied surface position error δP_2 resulting from the attitude error $\delta \theta$. The former is printed directly by the computer program. The latter may be computed by

$$\delta P_2 = 705 \text{ km} \sin \delta \theta \approx 705 \text{ km} \delta \theta.$$

It may be shown that the position error on the surface δP is bounded by

$$\{\delta P_1^2 + \delta P_2^2\}^{1/2} \leq \delta P \leq \delta P_1 + \delta P_2$$

Table IV-6 shows the upper bound of the boresight determination error for each update interval. The peak navigational position error is fairly uniform at approximately 7 meters. The peak attitude error seems to have a minimum around the 40-second update interval point. This suggests a minimum position error implied at the surface to result from a GCP update interval near 40 seconds. The higher attitude errors for longer intervals seem to arise from the effects of gyro drift. Speculation on the high attitude errors for very short update intervals suggests the cause may be interference or rectification resulting from the closeness of the update intervals of GCP and GPS. Because the total state is not observable using landmark tracker measurements alone, a short update interval causes artificially high covariance values.

2. Sensitivity to GCP Look Angles

Cases 9, 10, 12, and 14 examine the sensitivity of system performance to constant-amplitude look angle. The cases mentioned examine the effects of constant look angles of respectively 0, 2.5, 5, and 10 degrees with other conditions as previously specified and the GCP update interval set at 80 seconds. Cases 17 and 18 examine the sensitivity to look angles of uniform distribution between peaks of ± 2.5 and ± 5.0 degrees respectively again with an 80-second GCP update interval. All other conditions remained the same, and no star tracker measurements were used.

The cases 9, 10, 12 and 14 showed an insensitivity to the magnitude of constant look angle with what appears to be a very long convergence time. As previously discussed, this is probably due to lack of total observability in this measurement situation. With random look angles convergence is more rapid with a ± 5 -degree variation than with a ± 2.5 -degree variation. However a comparison of case 12 with case 5 shows virtually identical convergence characteristics. The explanation is not apparent at this time.

3. GPS Update Interval Sensitivity

Cases 21, 22, 23, and 24 examine the system performance sensitivity to the GPS update interval. The intervals used were respectively 10, 20, 40 and 80 seconds. The GCP look angle was maintained random at ± 10 degrees maximum, and the update interval was set to 80 seconds while all other parameters were held at their previous values.

The peak values of error for position, velocity and attitude are presented in Table IV-7. This table shows an apparent peak in position and attitude errors at or near the GPS update interval of 40 seconds, falling off on both sides. No explanation is readily available for this result except that possibly no 3σ numbers were encountered due to the small number of data points.

Table IV-7 Results of GPS Update Interval Sensitivity

Case	GPS Update Interval, s	Peak Position Error, m	Peak Attitude Error, s	Implied Surface Error Bounds, m
5	5	6.5	16	18-20
21	10	7.0	13	15-17
22	20	8.2	16	18-21
23	40	12.0	15	18-21
24	80	7.5	13	15-17

4. GPS Accuracy Sensitivity

To determine how sensitive the total system is to the accuracy of the GPS updates, the noise levels of the position and velocity as well as the filter knowledge of these components were increased first in velocity by a factor of 4 and run as case 25. Then the position components were increased by a factor of 4, restoring the position error to its original value. This was run as case 25. All other parameters were as previously specified.

The results show a definite impact on performance resulting from the higher noise levels in the GPS position and velocity channels. However, the increased size of the error in performance did not correspond directly with the increased measurement noise.

Neither attitude error nor its convergence characteristic were significantly affected by these changes in measurement noise or the resulting increased spacecraft navigational error. The projected error in position on the earth's surface will be increased by the additional spacecraft navigation position uncertainty in the manner previously discussed.

5. Effects of Clouds

Cloud conditions were simulated by picking a random number from a uniform population. This number yields a cloud density by interpolation from a density table. Should the cloud density exceed 40%, the system attempting a GCP update will default to one of the star trackers not occulted by Earth, moon, or sun. If the cloud density is between 10% and 40%, the GCP update is made but with degraded accuracy from 3 meters (normal) standard deviation on the surface to 30 meters standard deviation on the surface.

Case 30 examined the effects of clouds using the standard mechanization with a linear cloud density table from 0% to 90%. All other parameters were nominal, with GCP and GPS update intervals of 80 and 5 seconds, respectively.

The results are not greatly different from those previously seen except in attitude. It appears that the convergence rate is higher in the region from 0 to 300 seconds. This seems to result from the default to the star trackers occurring in the region of 400 seconds. The attitude update with star trackers is not coupled to the position, thereby resulting in faster convergence.

Near 1100 seconds on the attitude error curve, there is some increased attitude error. This resulted from cloud cover between 10 to 40%, forcing the higher GCP error referred to. The results did reconverge to acceptable values quickly.

6. GCP Recovery With and Without Star Tracker Backup

Case 31 simulated the effects of long periods of time without GCP updates. Here the landmark tracker was turned off and the system allowed to default to the star trackers for attitude update. Case 32 is similar to 31 although the star trackers were not available for update. The period of shutdown for both cases was approximately 1000 seconds after the system converged (2000 seconds) from its startup transient, 2000 to 3000 seconds. All other system parameters were held at nominal with a GPS updates at 5 seconds and GCP updates when active at 80 seconds.

In both cases the attitude error tends to grow in the period of no landmark tracker updates. However the bound on attitude error in case 31 is held down by the star tracker. Each case converges rapidly once the landmark tracker updates are again made active.

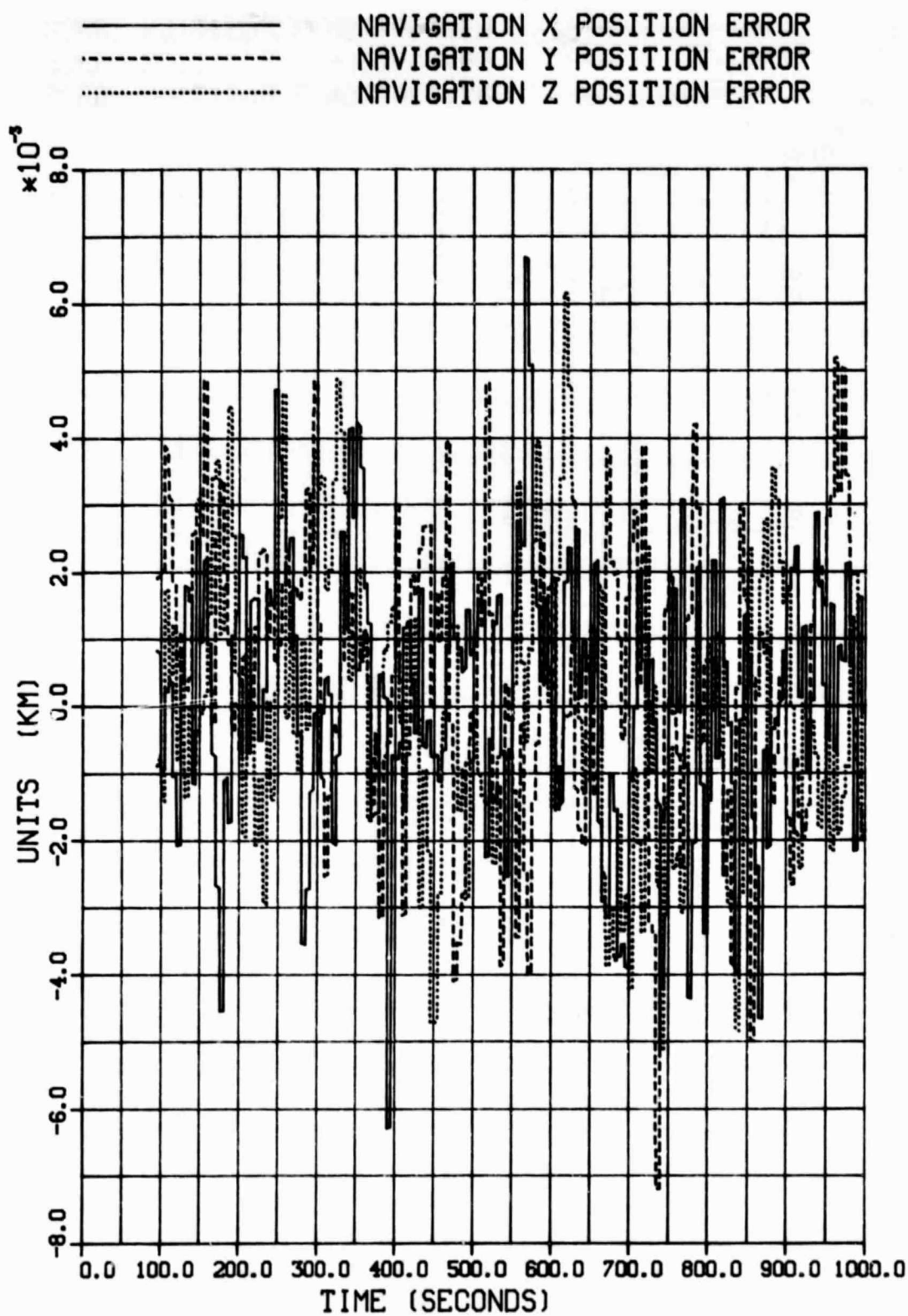
C. OVERALL CONCLUSIONS

The system as simulated met the requirements set down earlier for conditions where the GCP update intervals are between 20 and 80 seconds with GPS updates taken at about 10-second intervals. Although the bounds appear marginal (12 to 14 meters, Table IV-5), data are taken close to the starting time by comparison with some systems and the mission duration. In some cases part of the initial transient was included in estimation of the peak or 3 σ value for a given parameter. Therefore most of the results are felt to be pessimistic with respect to the size of the error bounds. These results also do not reflect the estimation of such gyro parameters as gyro bias, misalignment, scale factor error and nonorthogonality. With estimation and compensation for these parameters, the results should be further improved.

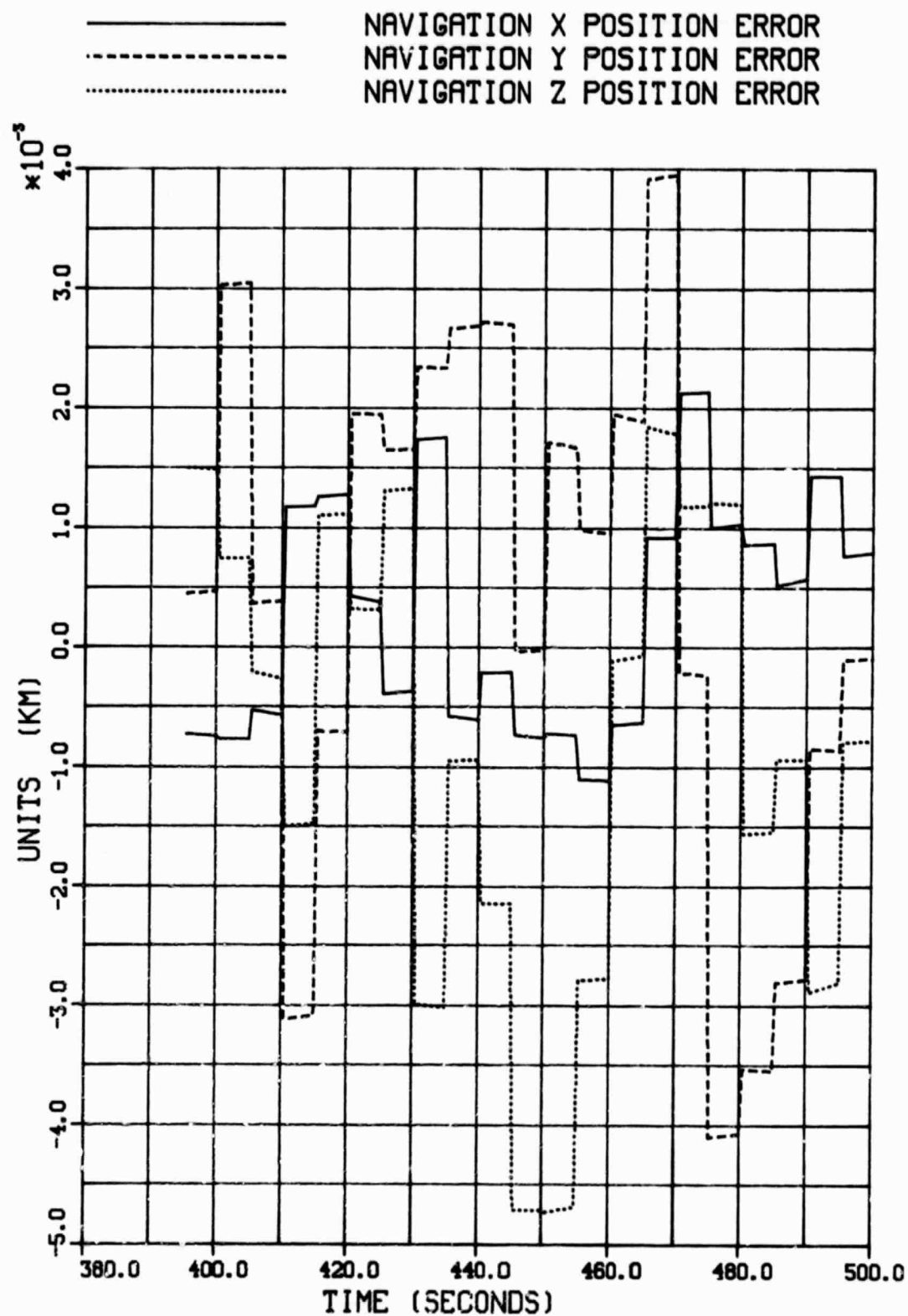
D. RECOMMENDATIONS

These studies took advantage of the fact that the reference body coordinate system is defined within the landmark tracker or GCP sensor. Therefore the misalignment of that sensor is by definition equal to zero. However, this shifts the error contribution of misalignment of other sensors if such misalignments are not estimated and compensated. The effects of such uncompensated terms as gyro bias may be seen in cases where long GCP update intervals are used. Further studies should be undertaken to implement the estimation and compensation of constant gyro parameters and possibly the landmark tracker parameters of bias and misalignment should that sensor coordinate frame not be taken as a body reference.

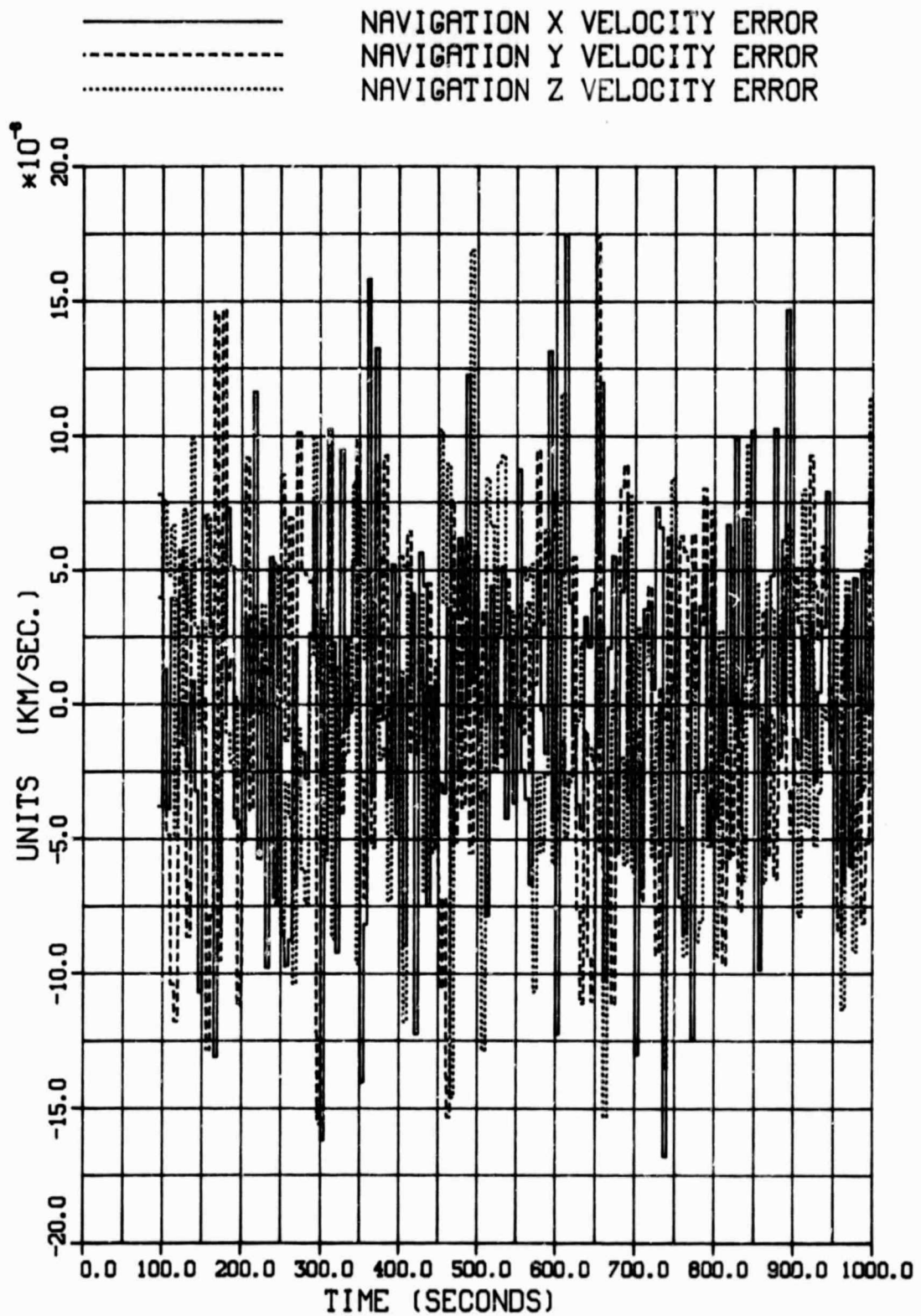
CASE1-DATA GCP-5SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



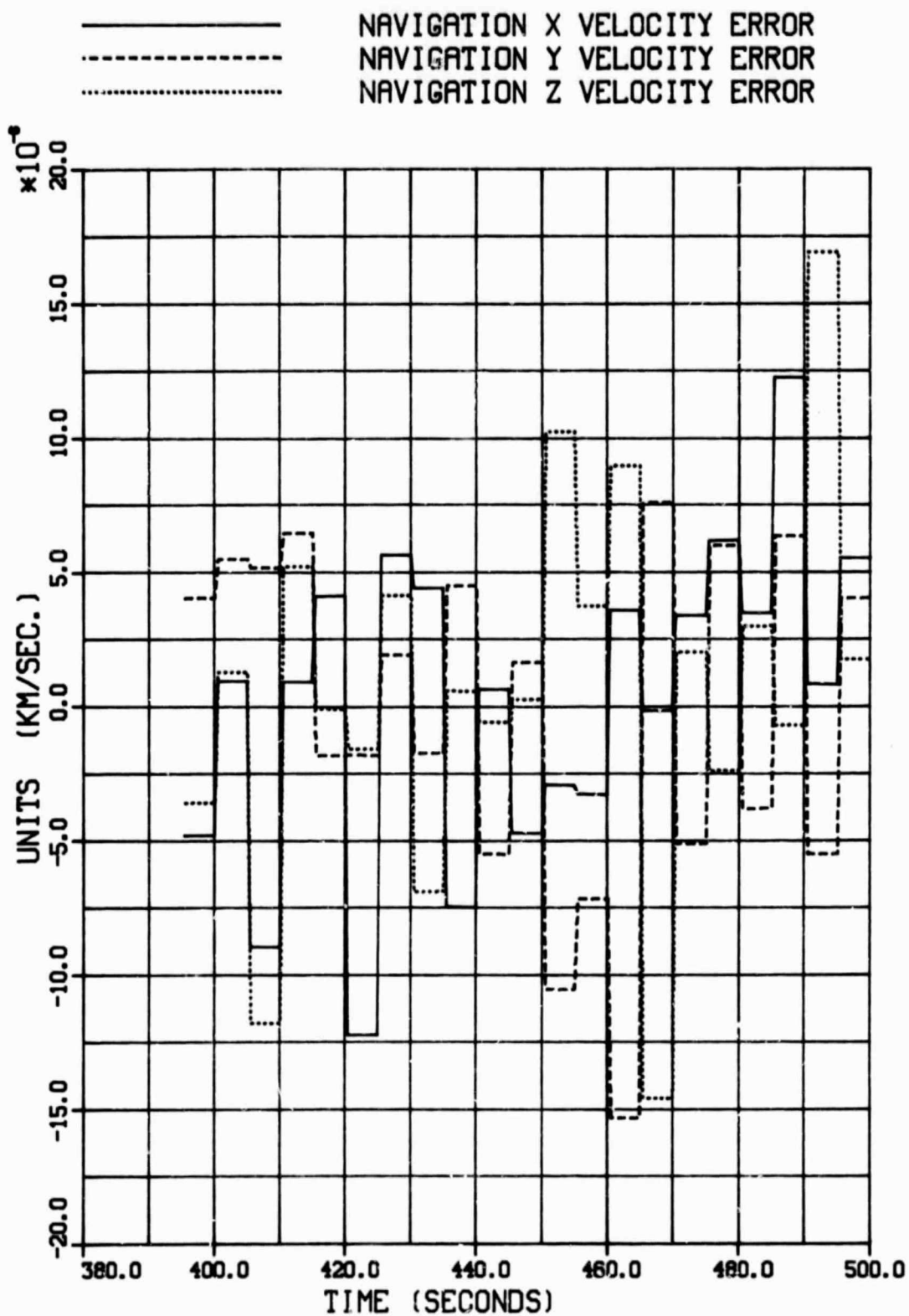
CASE1-DATA GCP-5SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



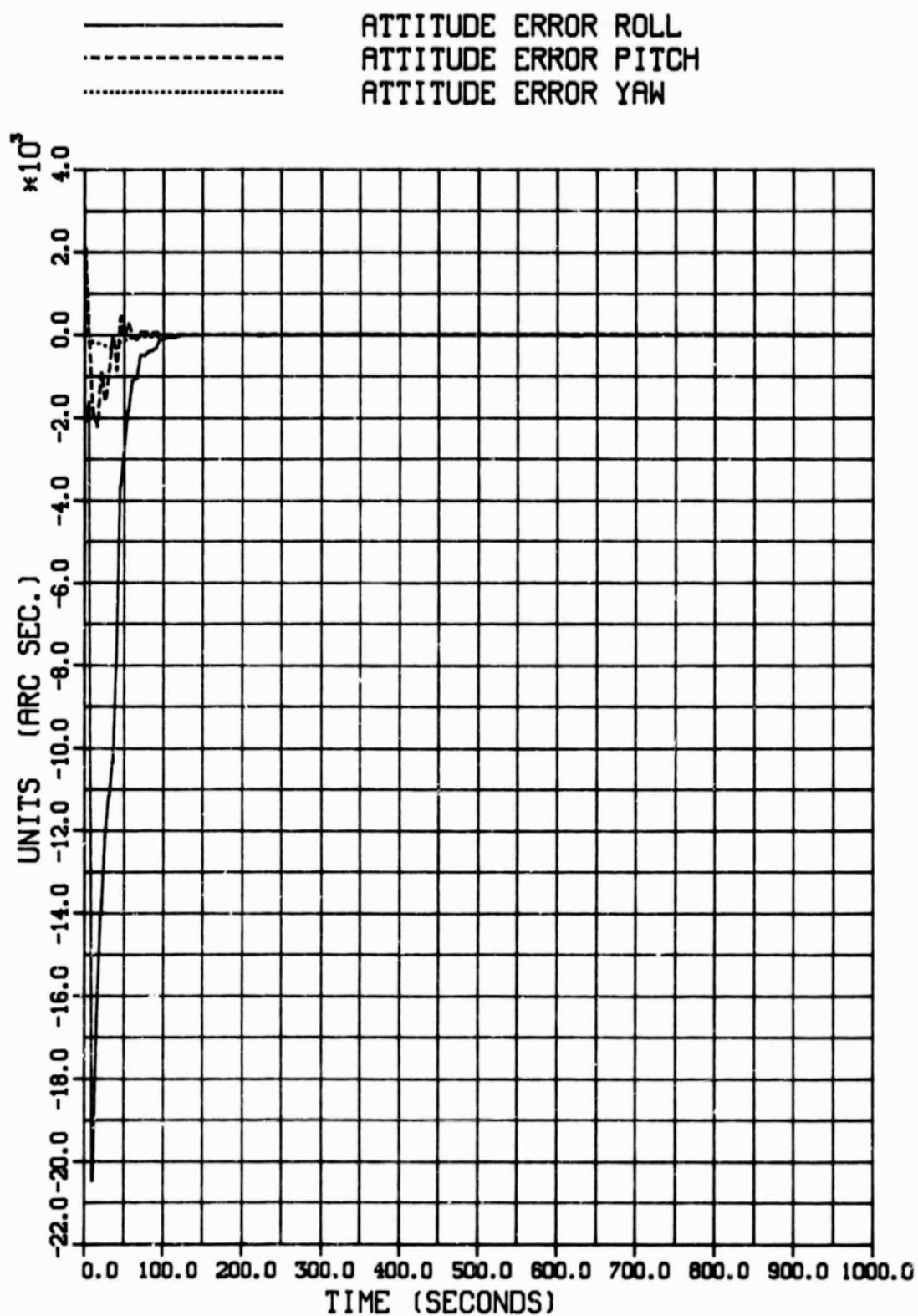
CASE1-DATA GCP-5SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***



CASE1-DATA GCP-5SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***

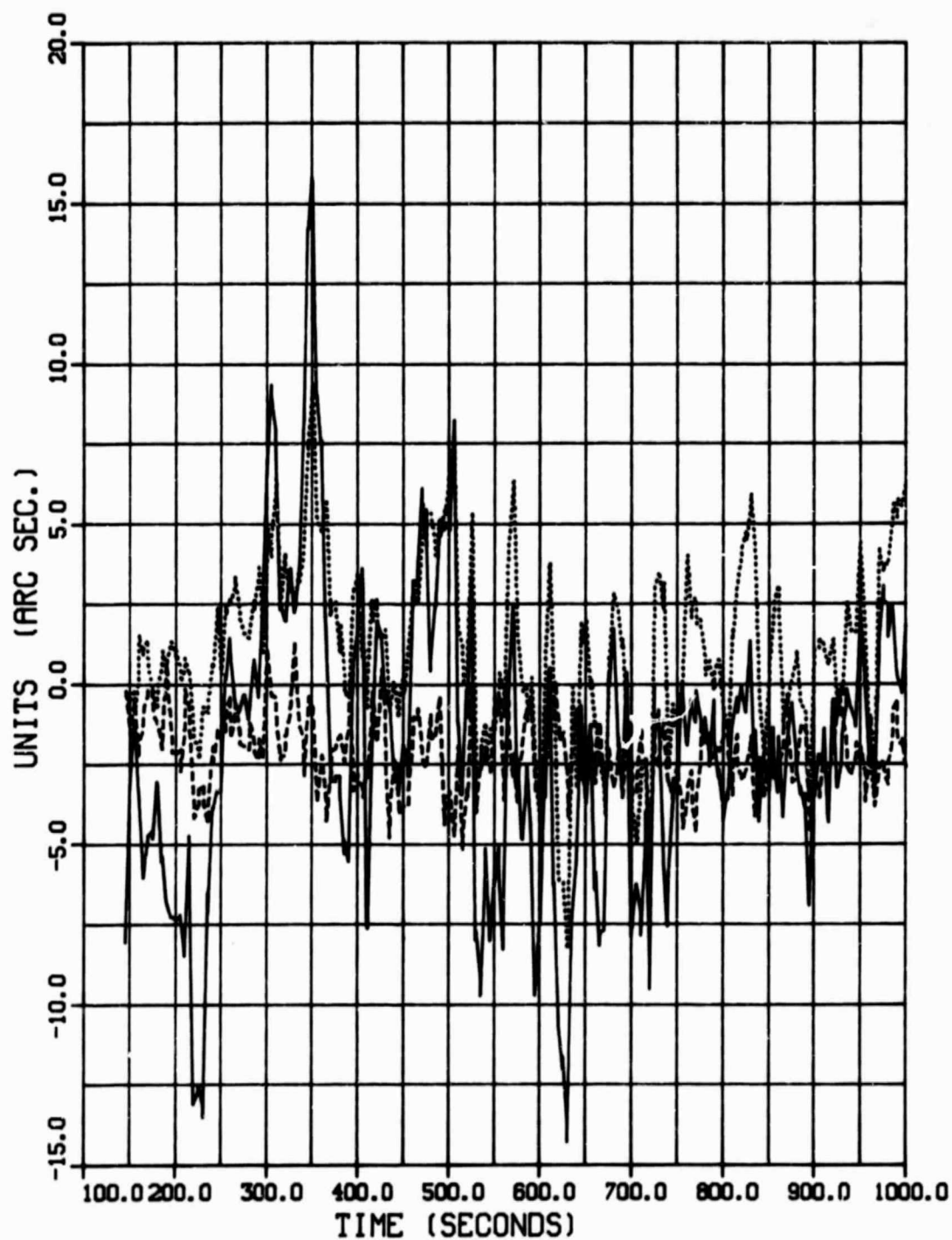


CASE1-DATA GCP-5SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

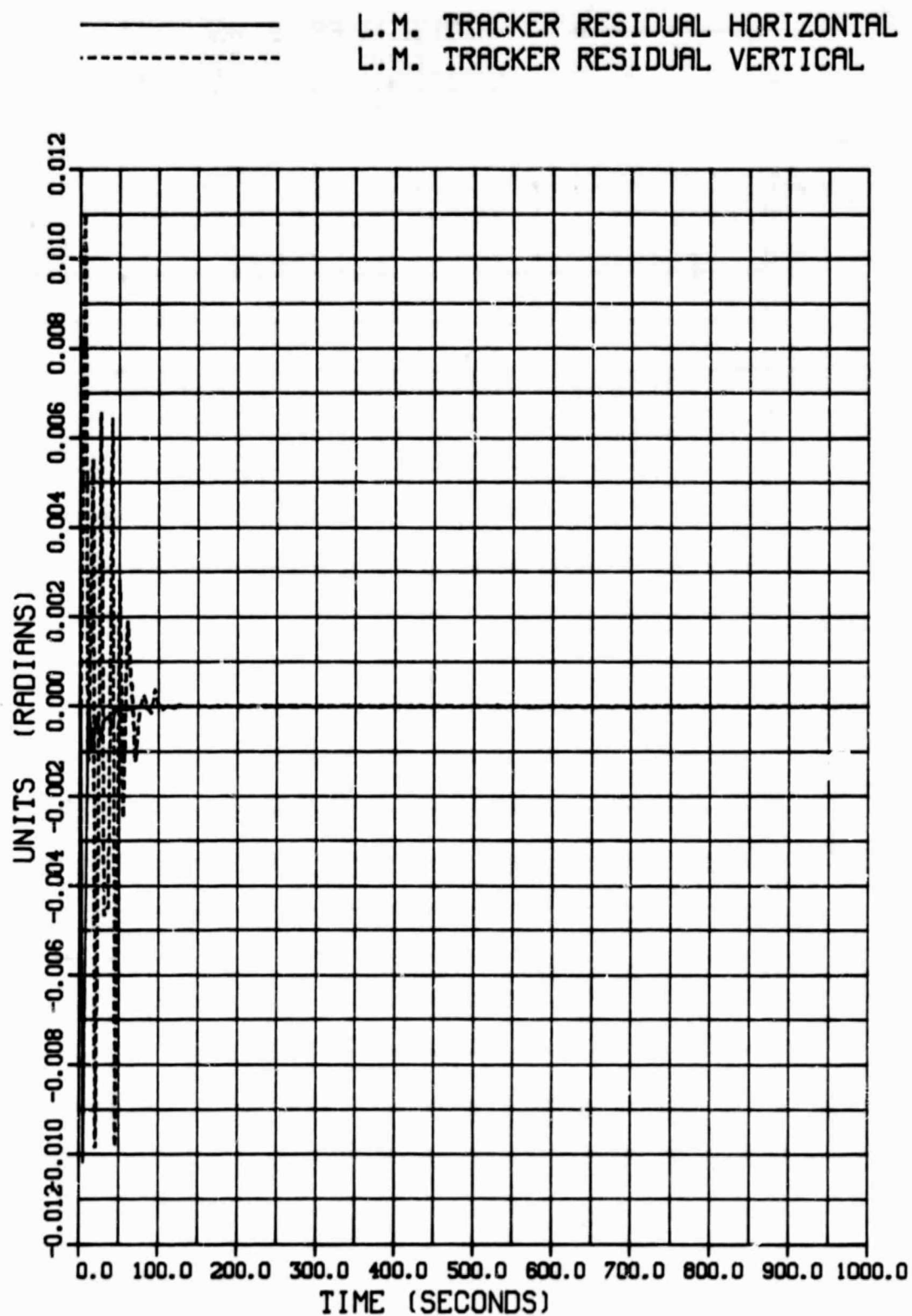


CASE1-DATA GCP-5SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

—— ATTITUDE ERROR ROLL
- - - - ATTITUDE ERROR PITCH
..... ATTITUDE ERROR YAW

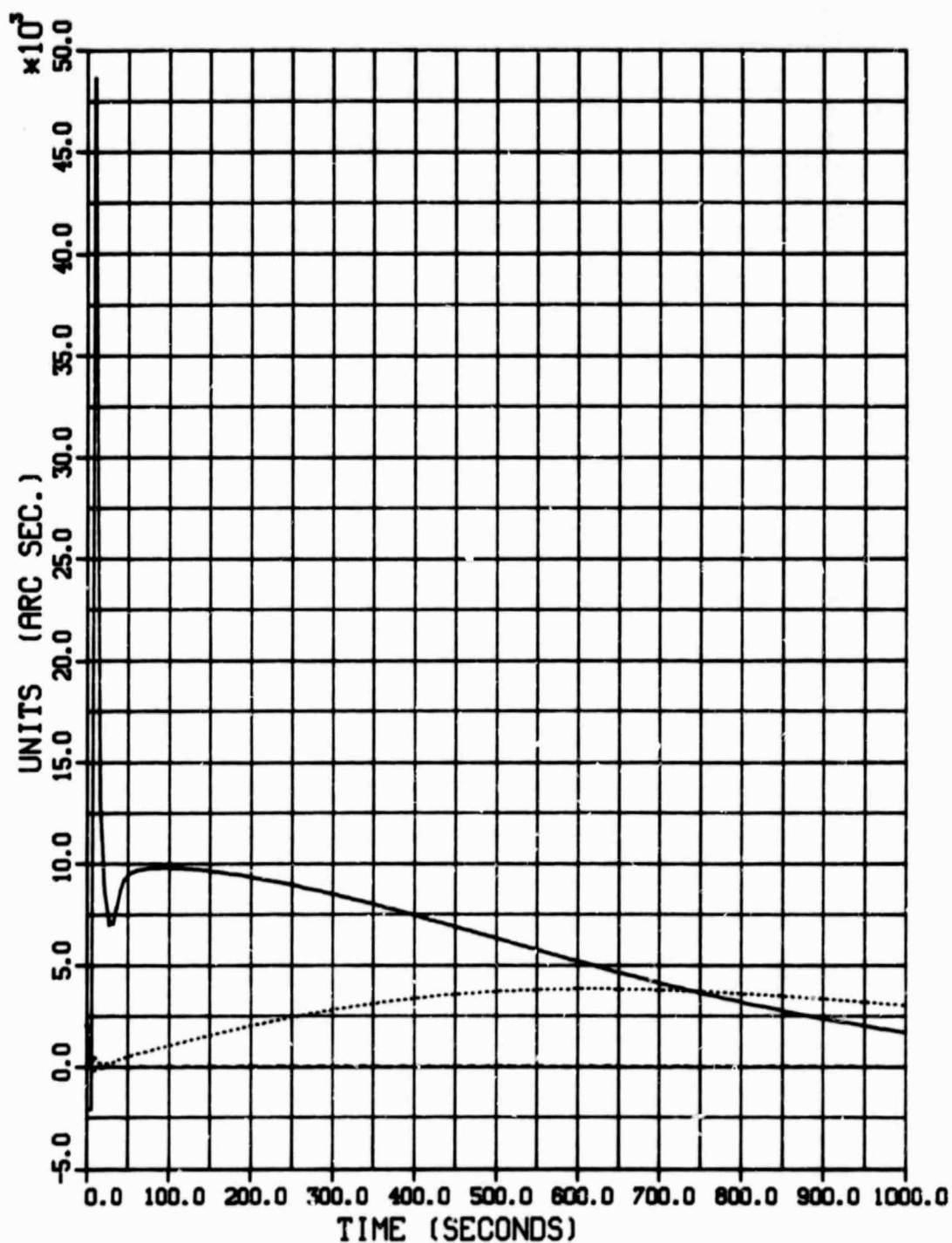


CASE1-DATA GCP-5SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

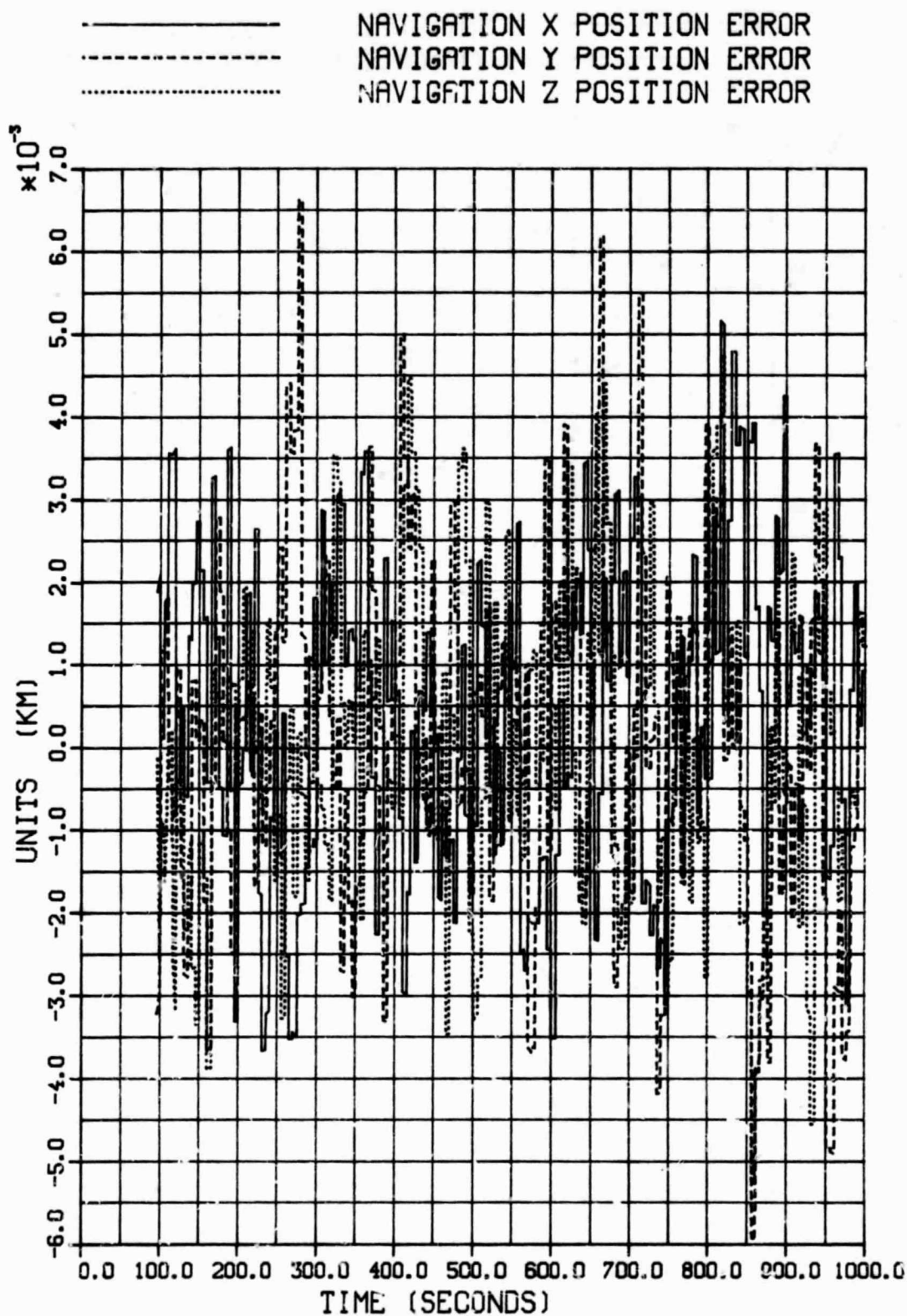


CASE1-DATA GCP-5SEC GPS-5SEC LA - FIXED
 *** INERTIAL COORDINATE FRAME ***

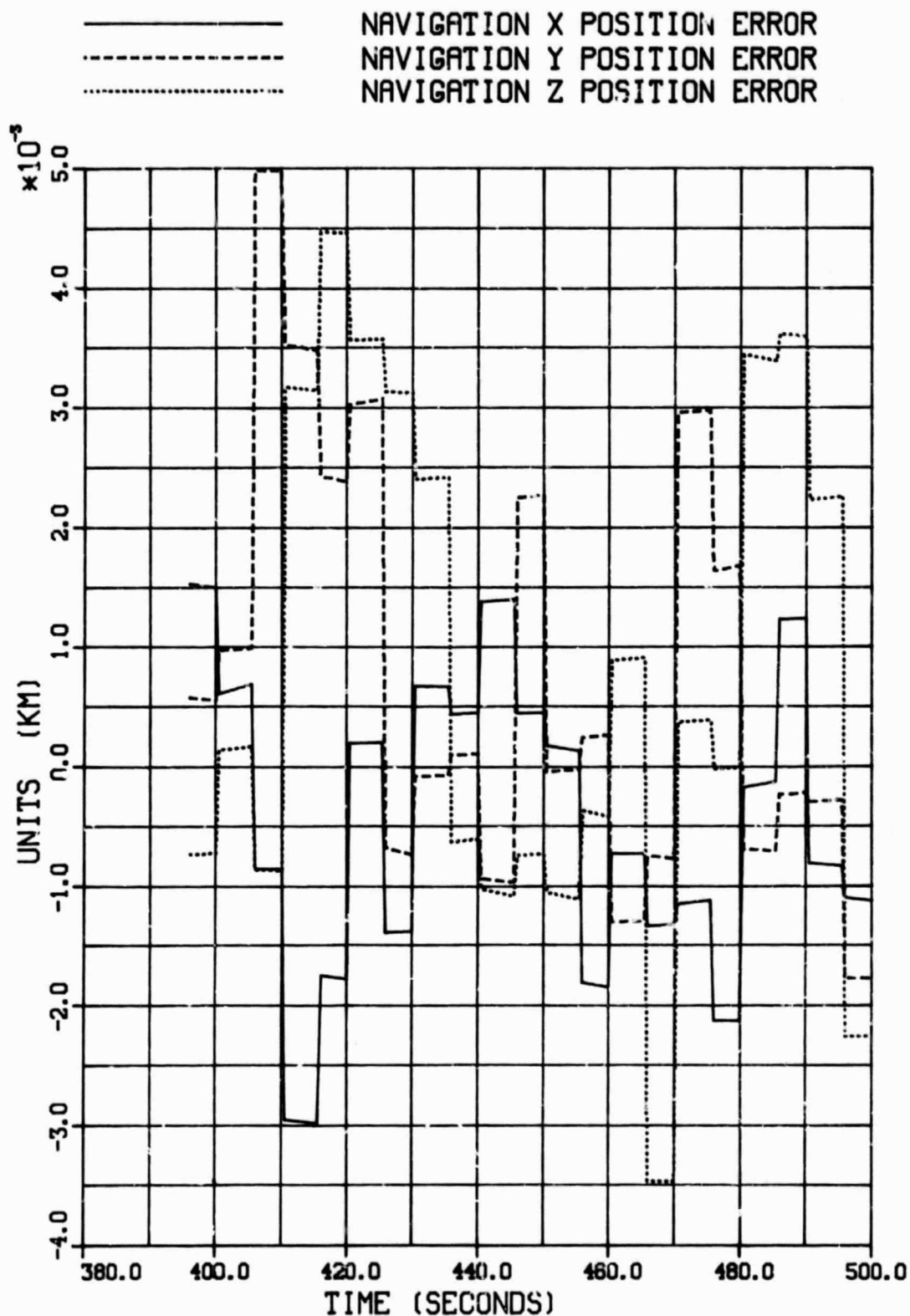
———— ATTITUDE ERROR ROLL
 - - - - - ATTITUDE ERROR PITCH
 ATTITUDE ERROR YAW



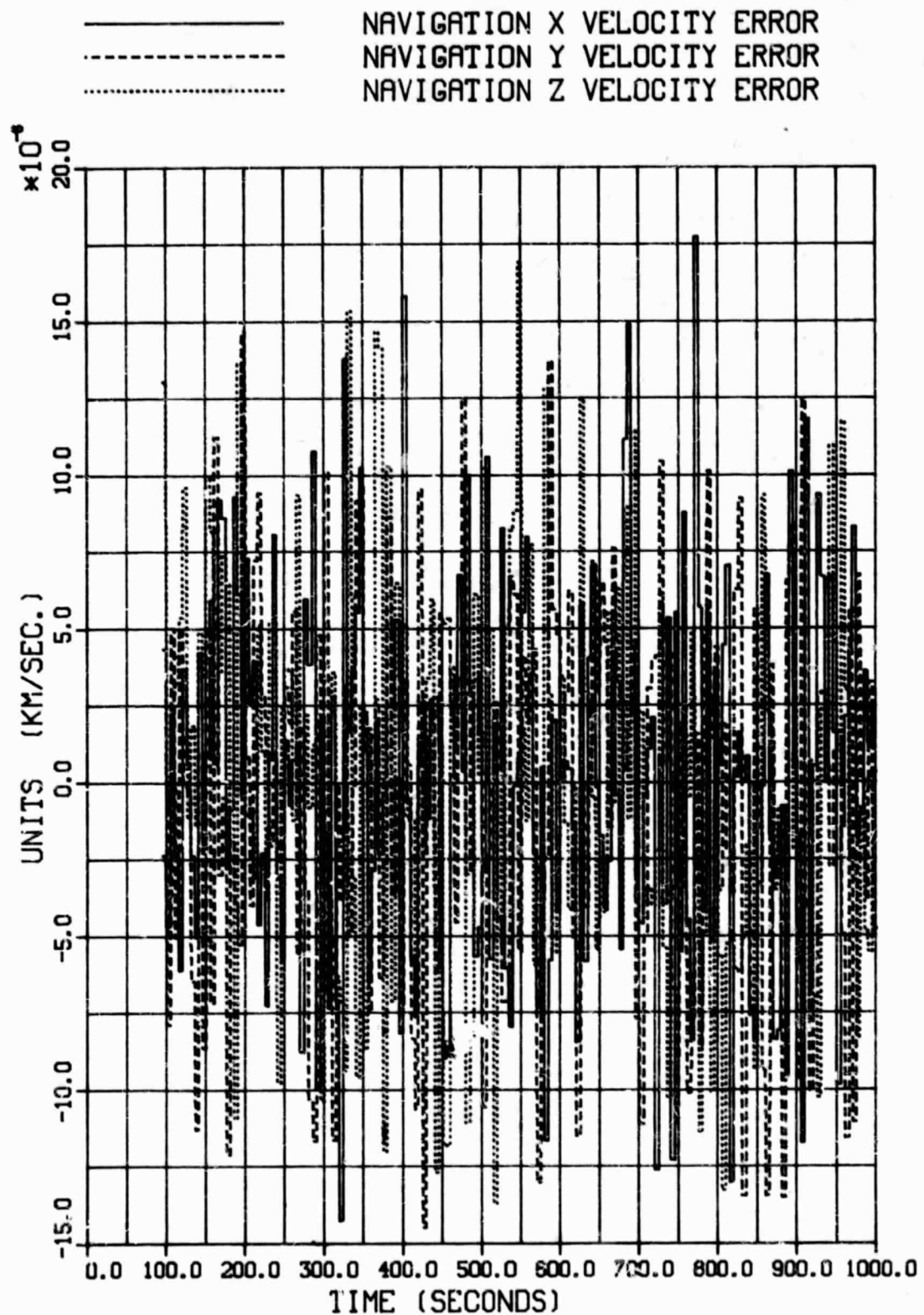
CASE2-DATA GCP-10SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***



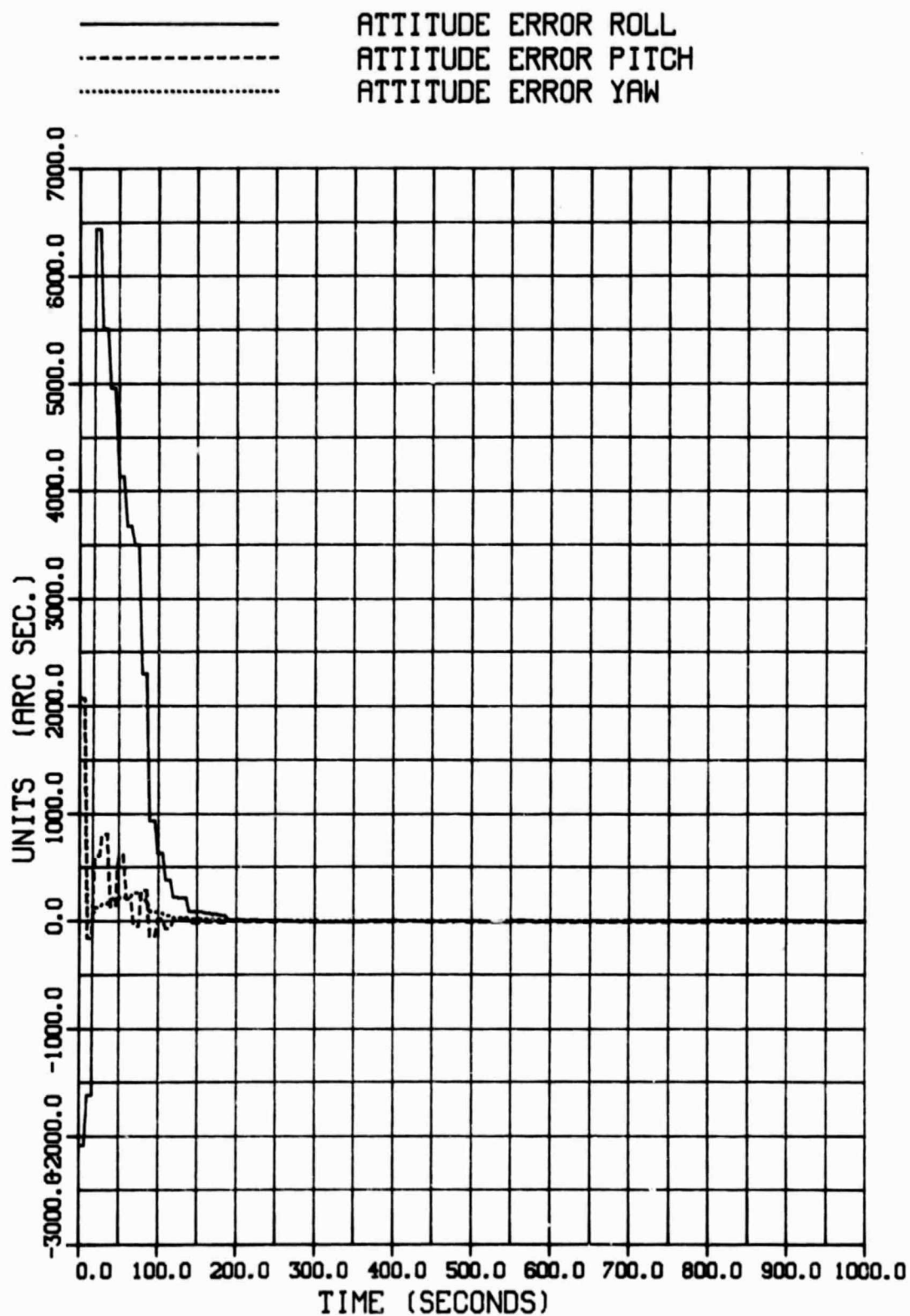
CASE2-DATA GCP-10SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***



CASE2-DATA GCP-10SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***

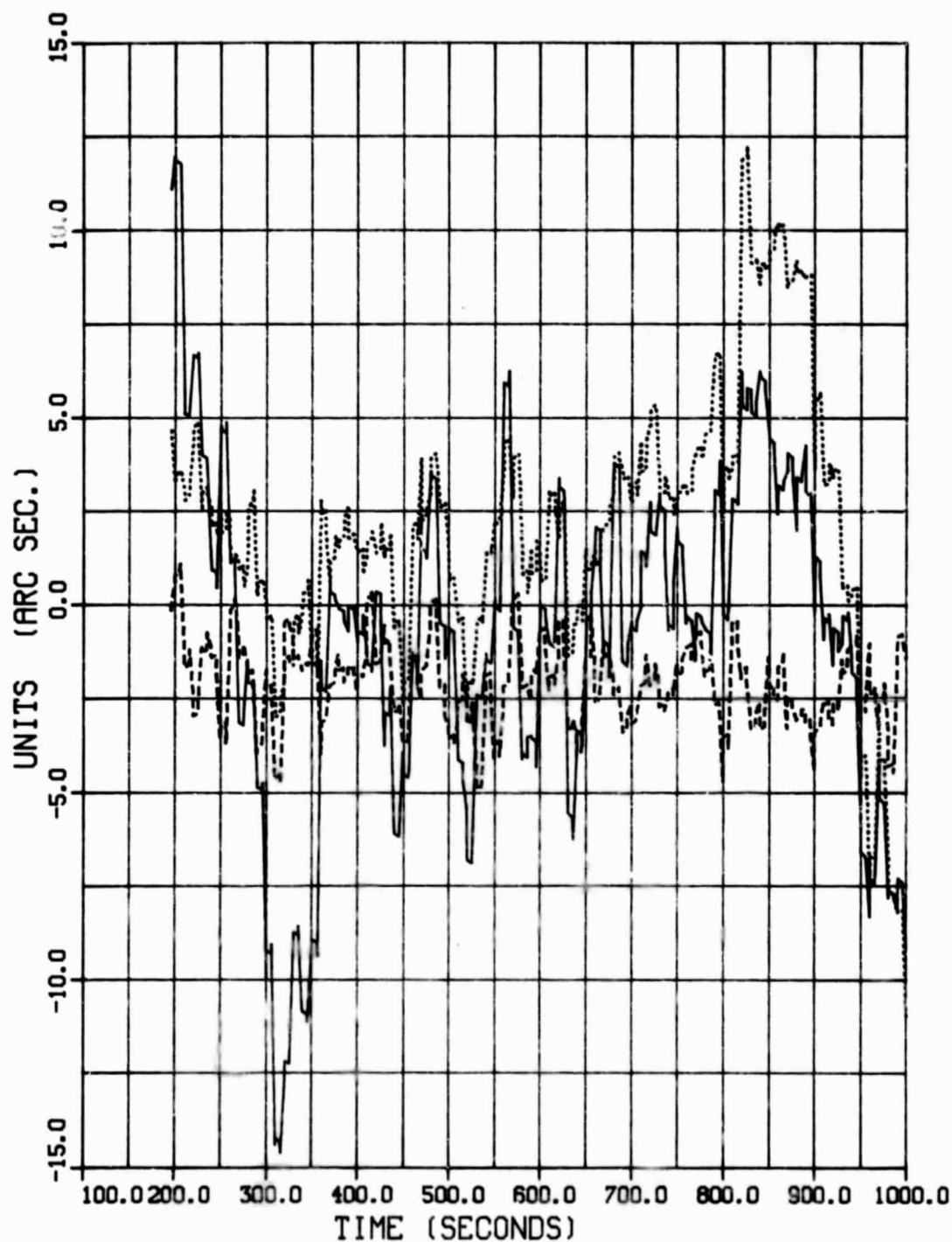


CASE2-DATA GCP-10SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



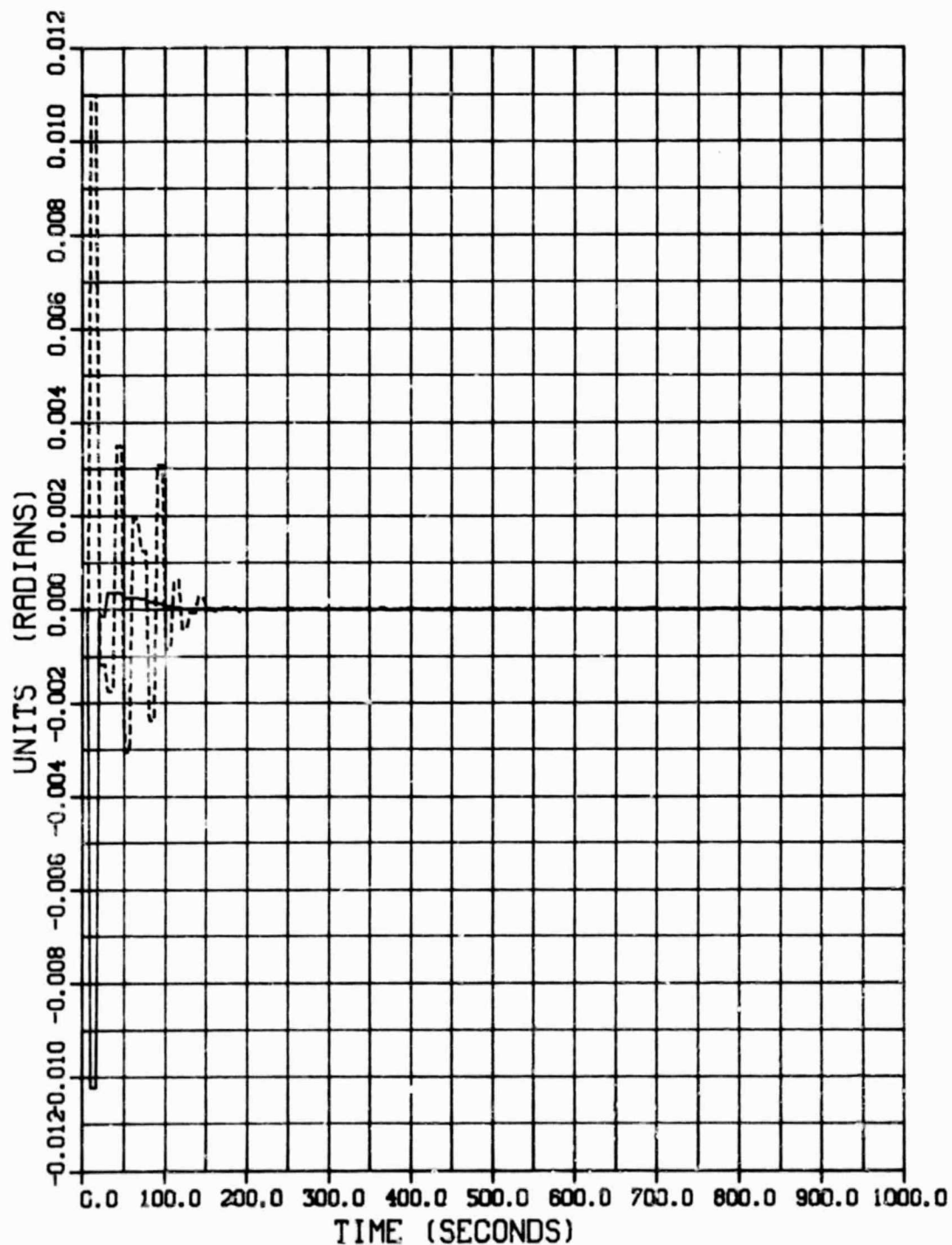
CASE2-DATA GCP-10SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

— ATTITUDE ERROR ROLL
- - - ATTITUDE ERROR PITCH
... ATTITUDE ERROR YAW



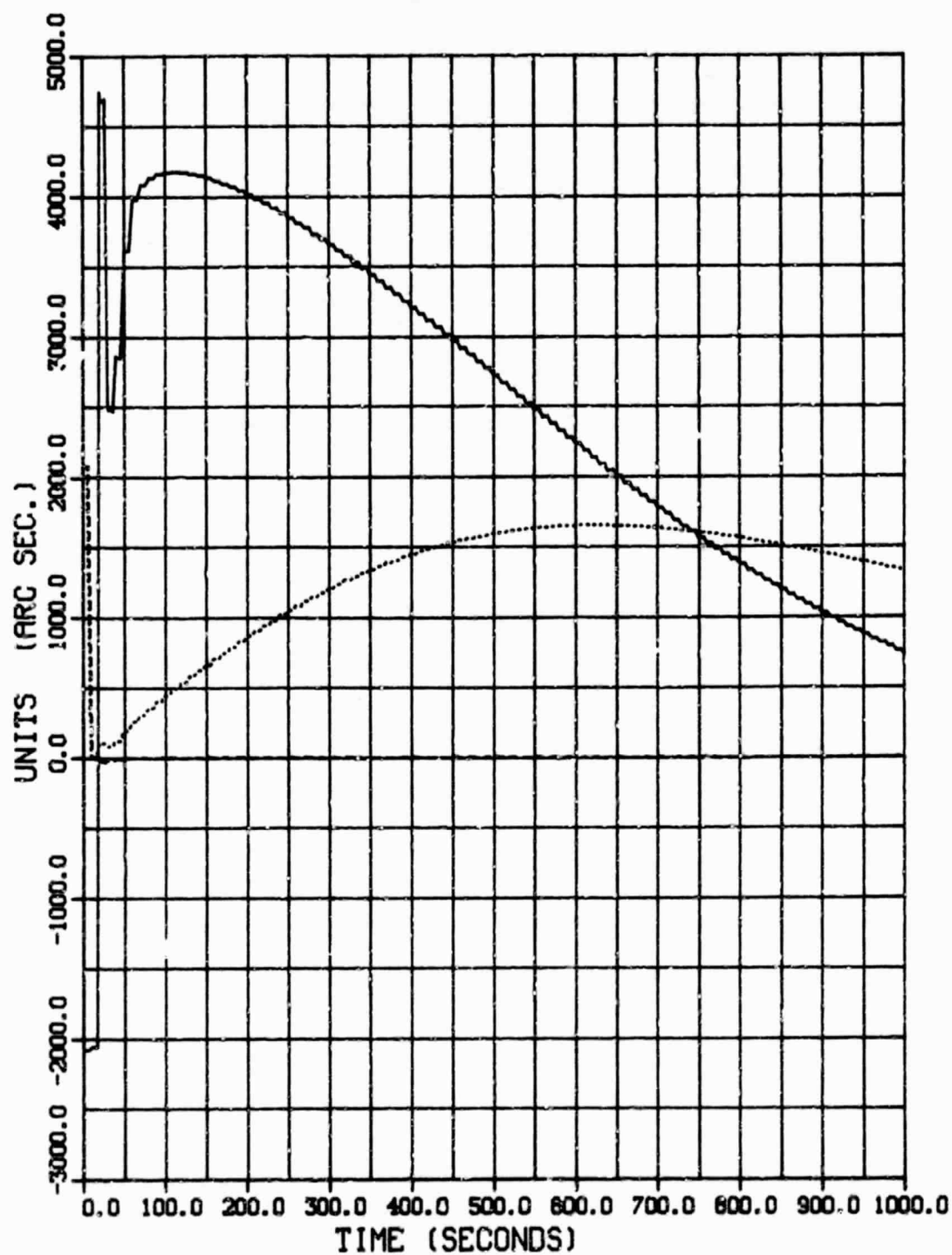
CASE2-DATA GCP-10SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

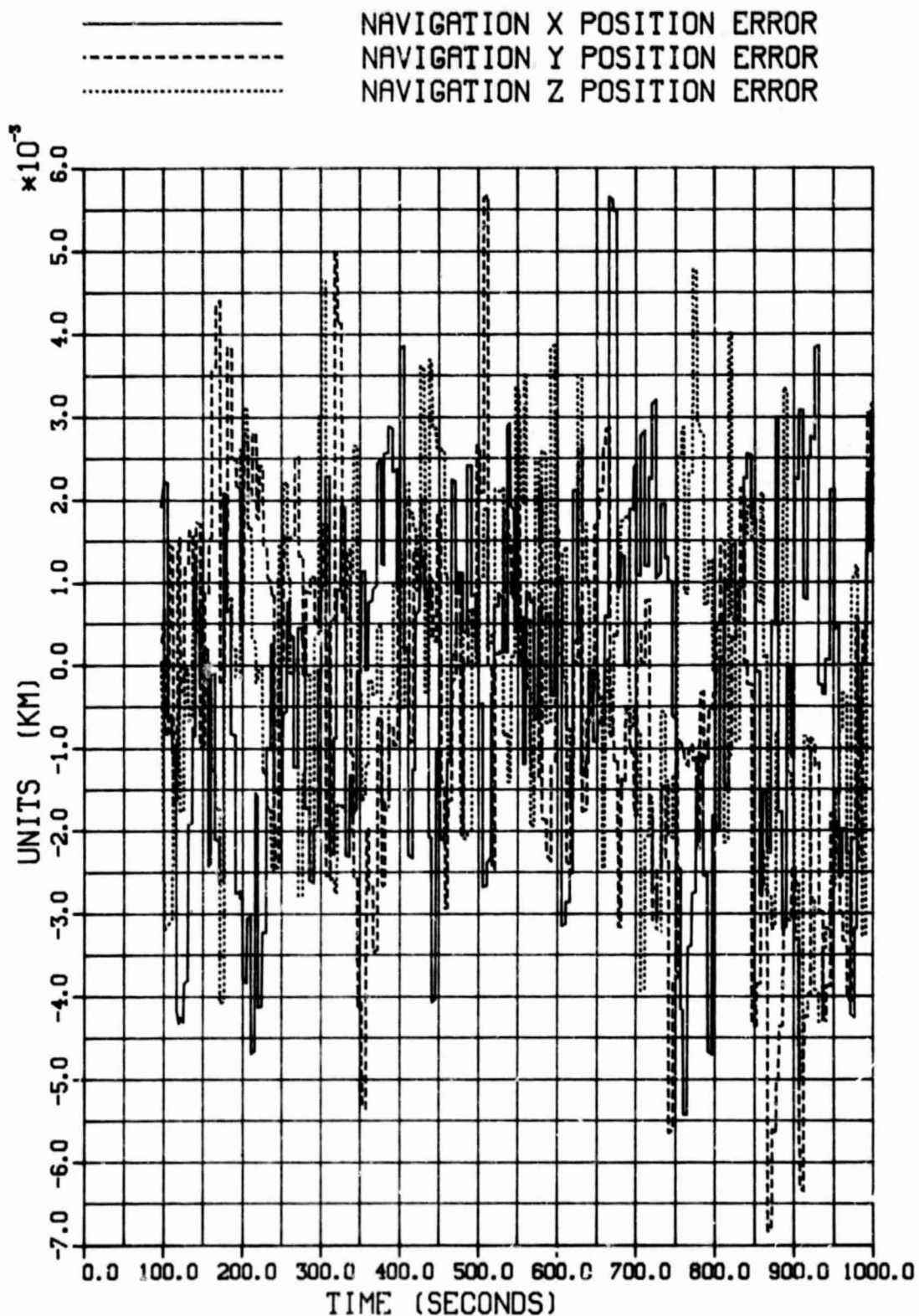


CASE2-DATA GCP-10SEC GPS-5SEC LA - FIXED
*** INERTIAL COORDINATE FRAME ***

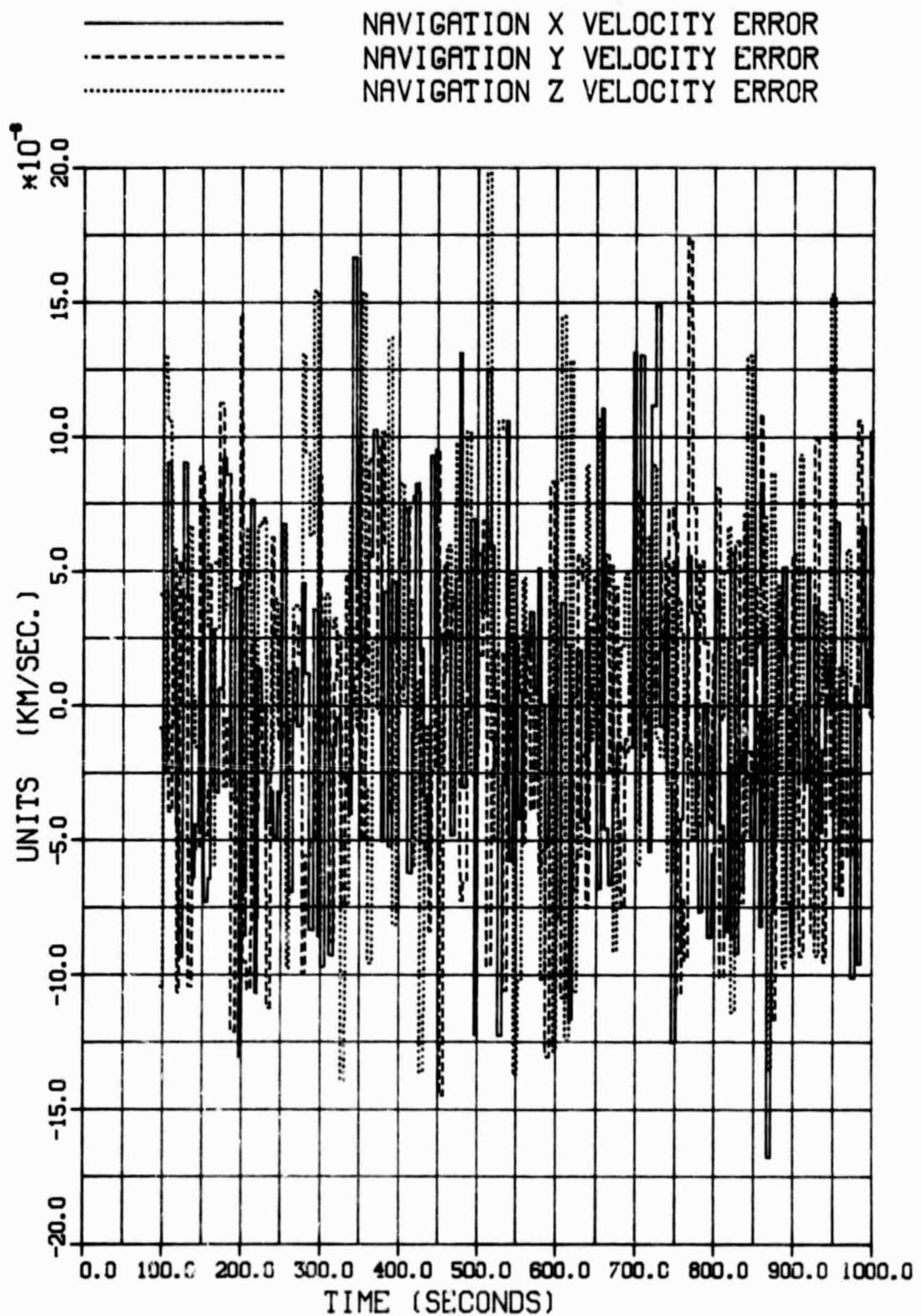
————— ATTITUDE ERROR ROLL
----- ATTITUDE ERROR PITCH
..... ATTITUDE ERROR YAW



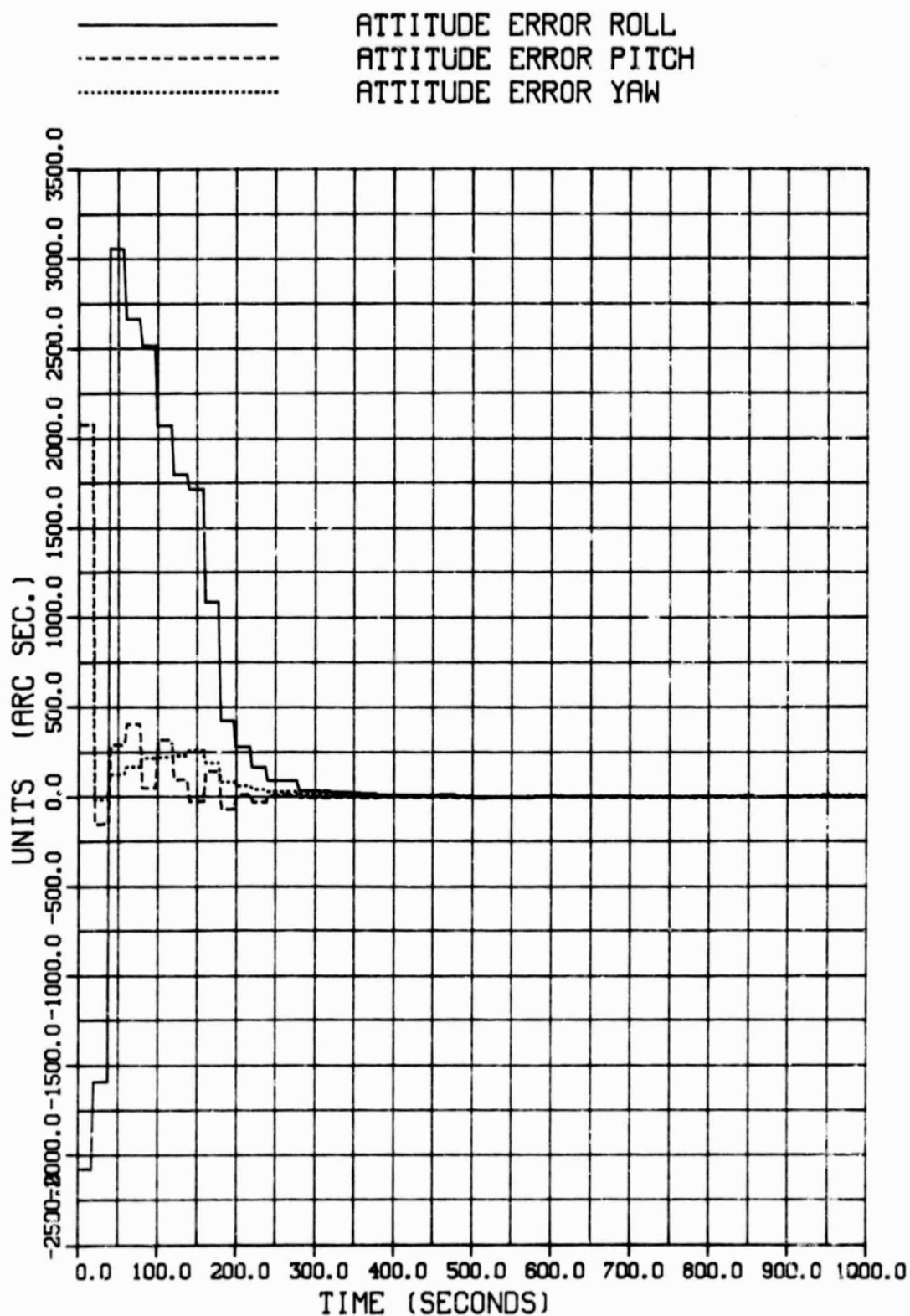
CASE3-DATA GCP-20SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



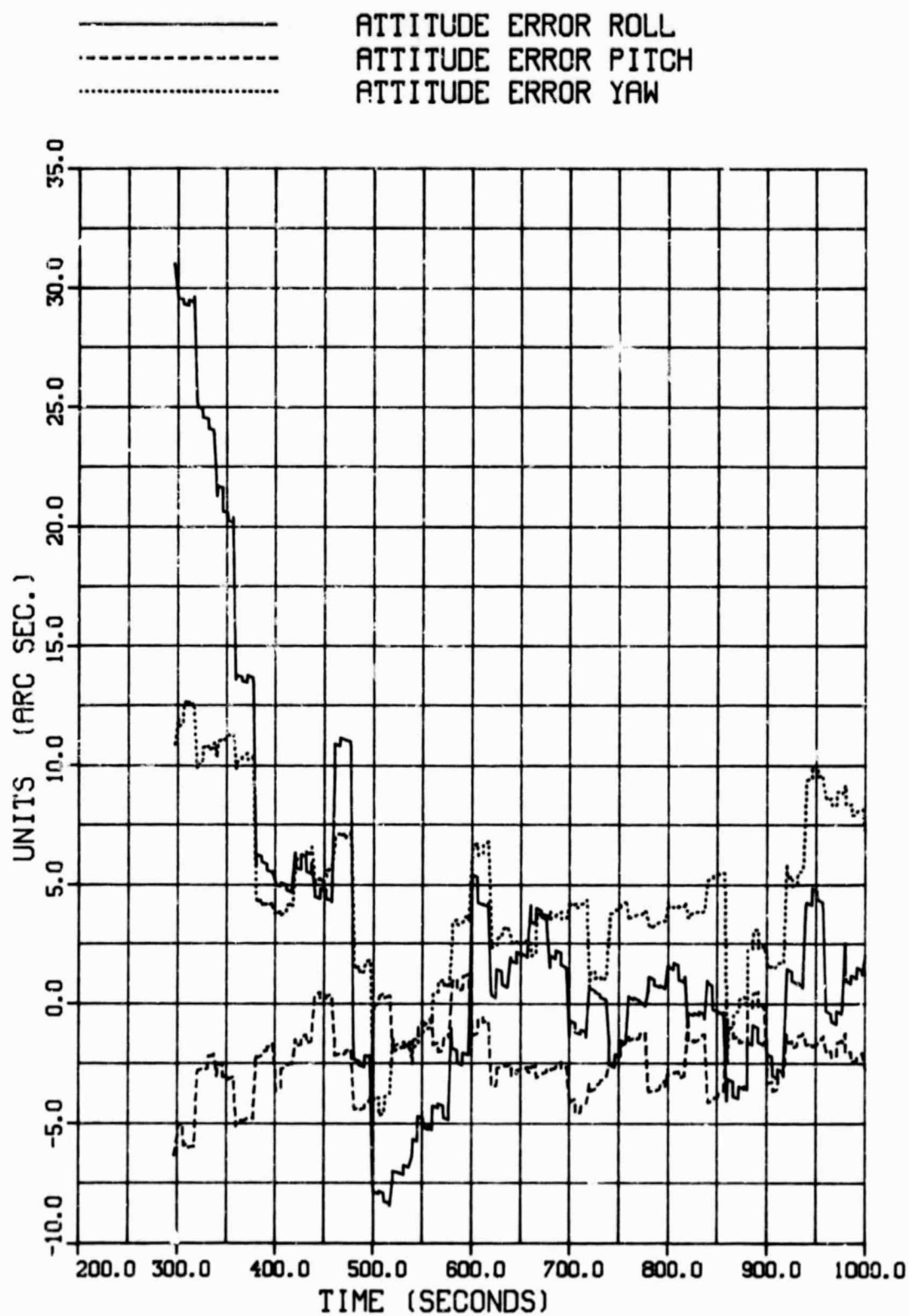
CASE3-DATA GCP-20SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***



CASE3-DATA GCP-20SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



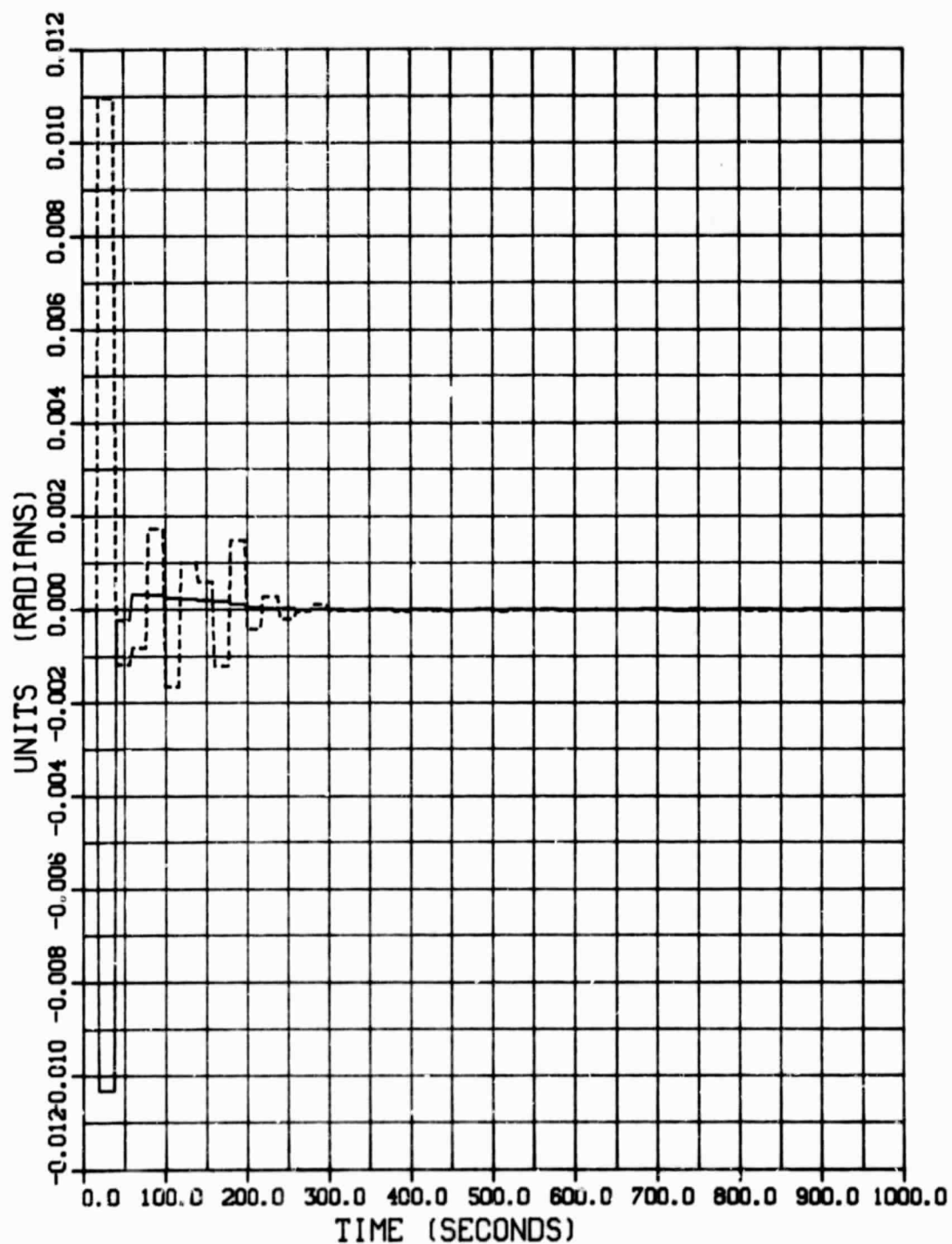
CASE3-DATA GCP-20SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



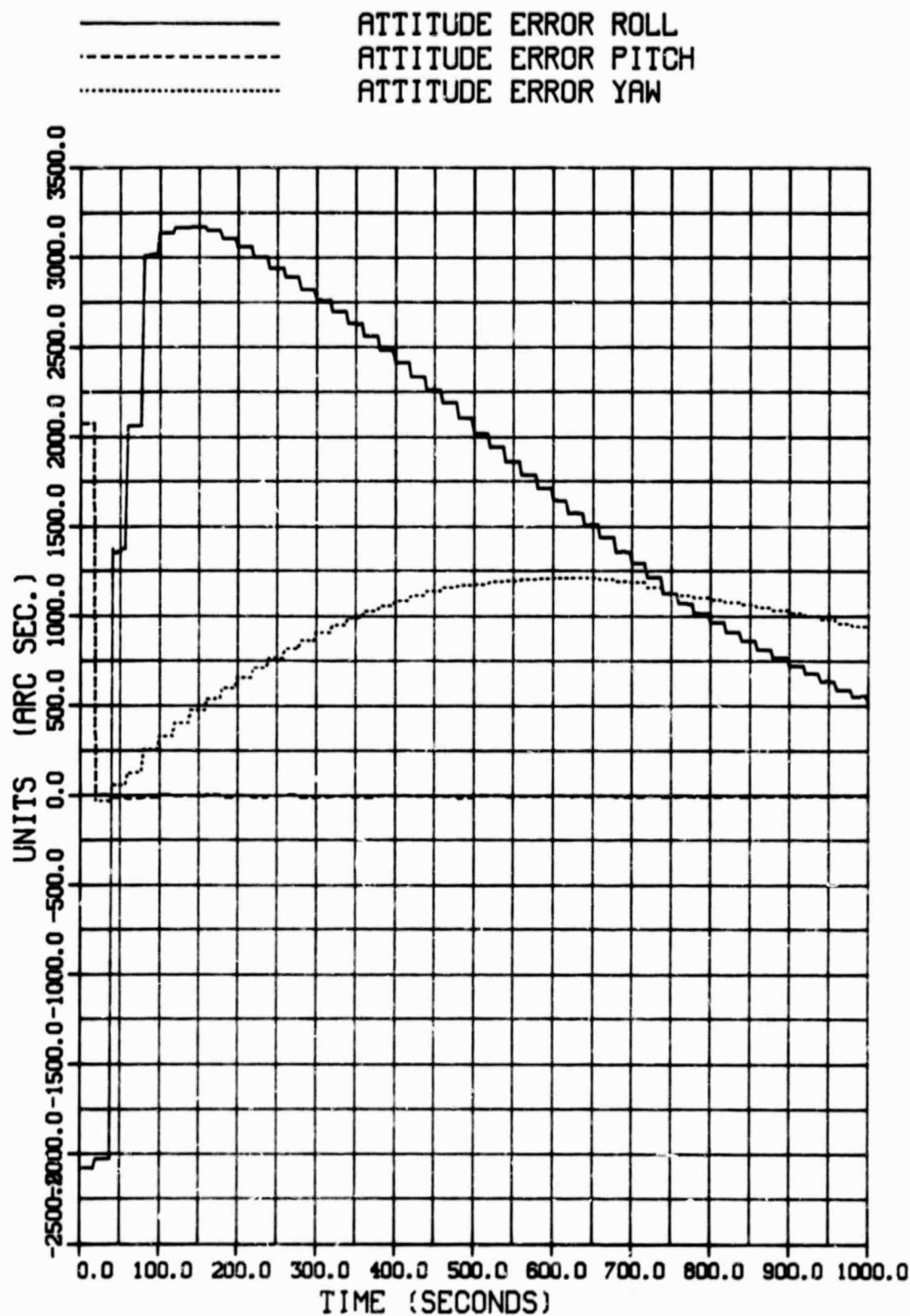
CASE3-DATA GCP-20SEC GPS-5SEC

*** INERTIAL COORDINATE FRAME ***

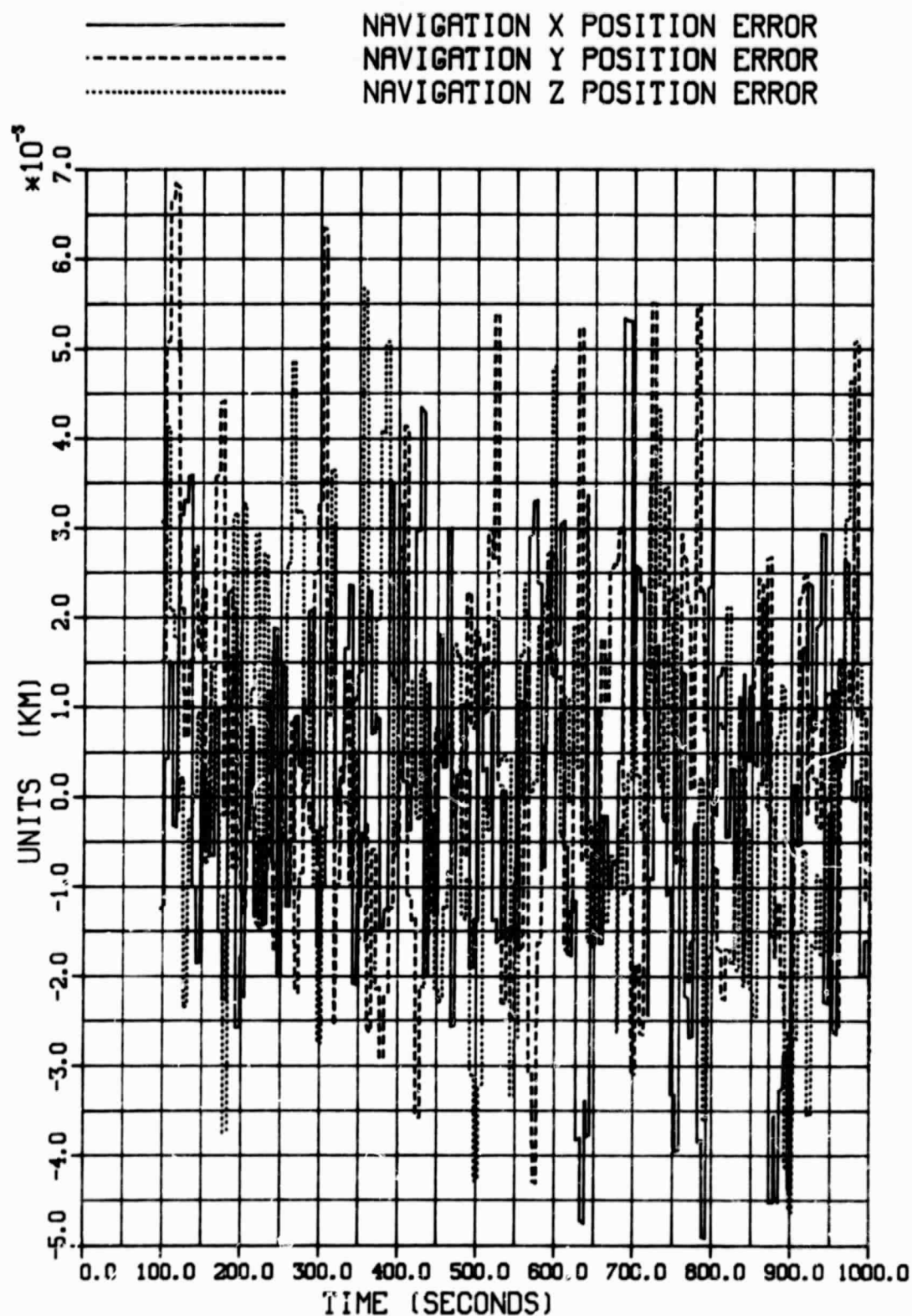
———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL



CASE3-DATA GCP-20SEC GPS-5SEC LA - FIXED
*** INERTIAL COORDINATE FRAME ***

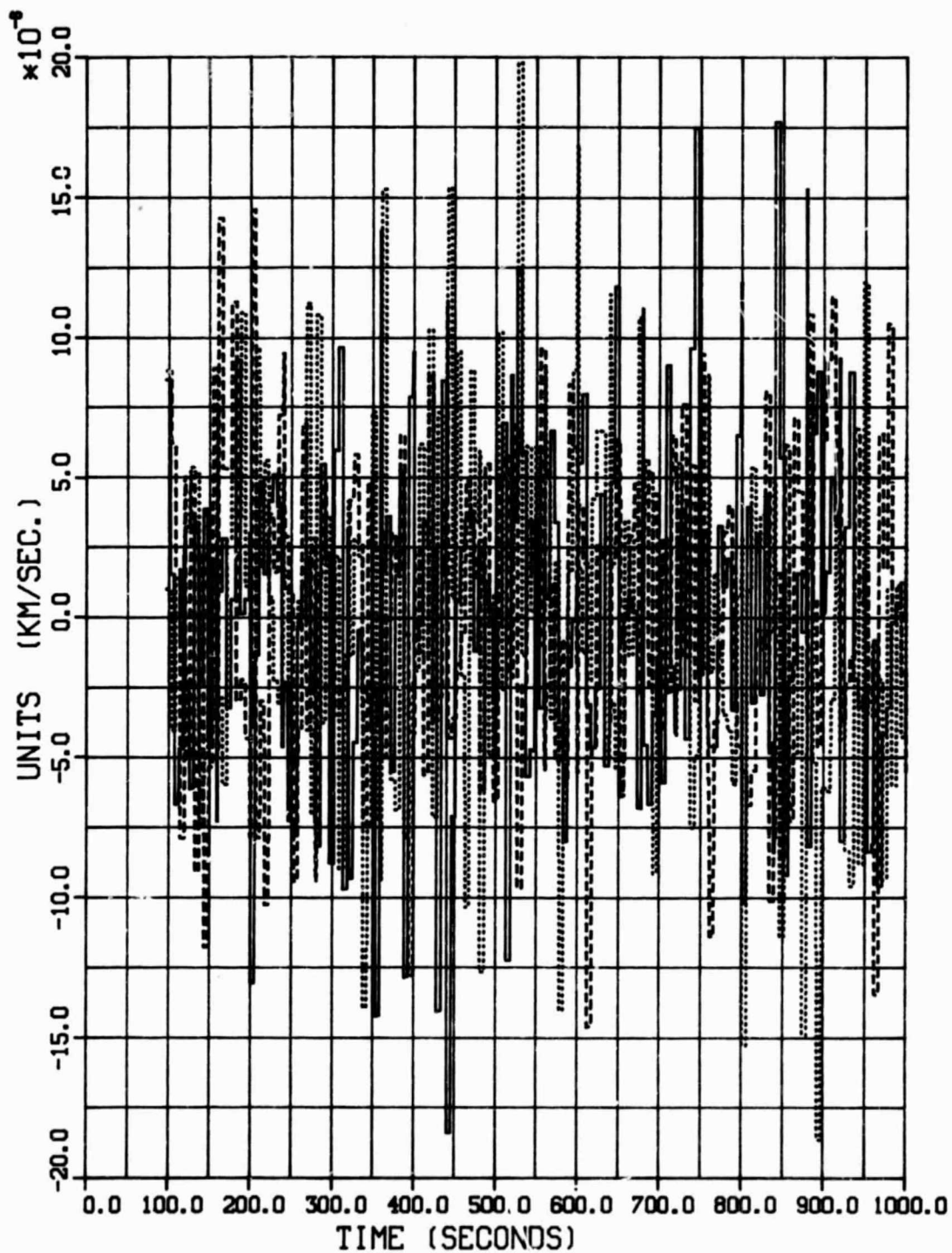


CASE4-DATA GCP-40SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



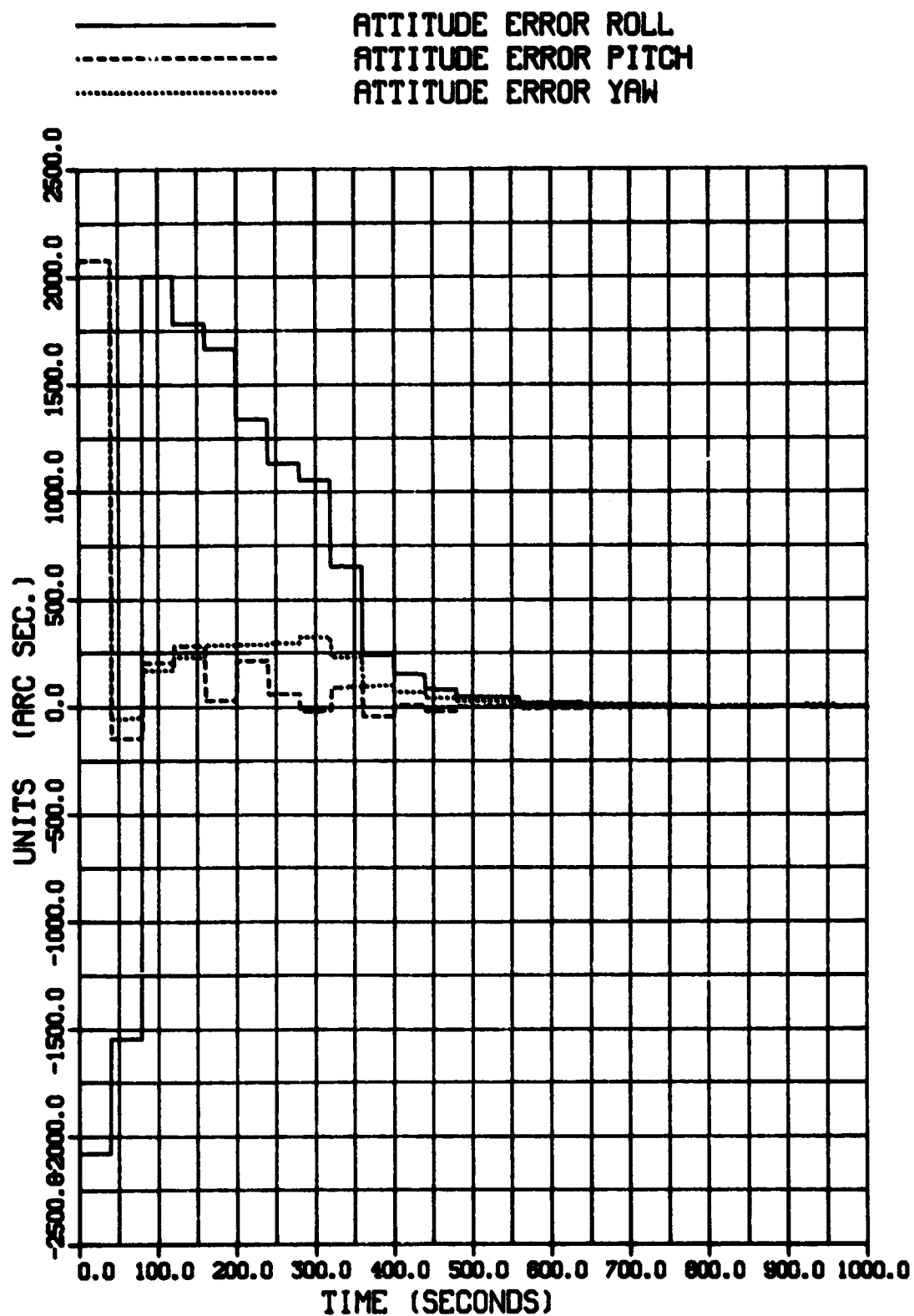
CASE4-DATA GCP-40SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***

—— NAVIGATION X VELOCITY ERROR
 - - - NAVIGATION Y VELOCITY ERROR
 . . . NAVIGATION Z VELOCITY ERROR

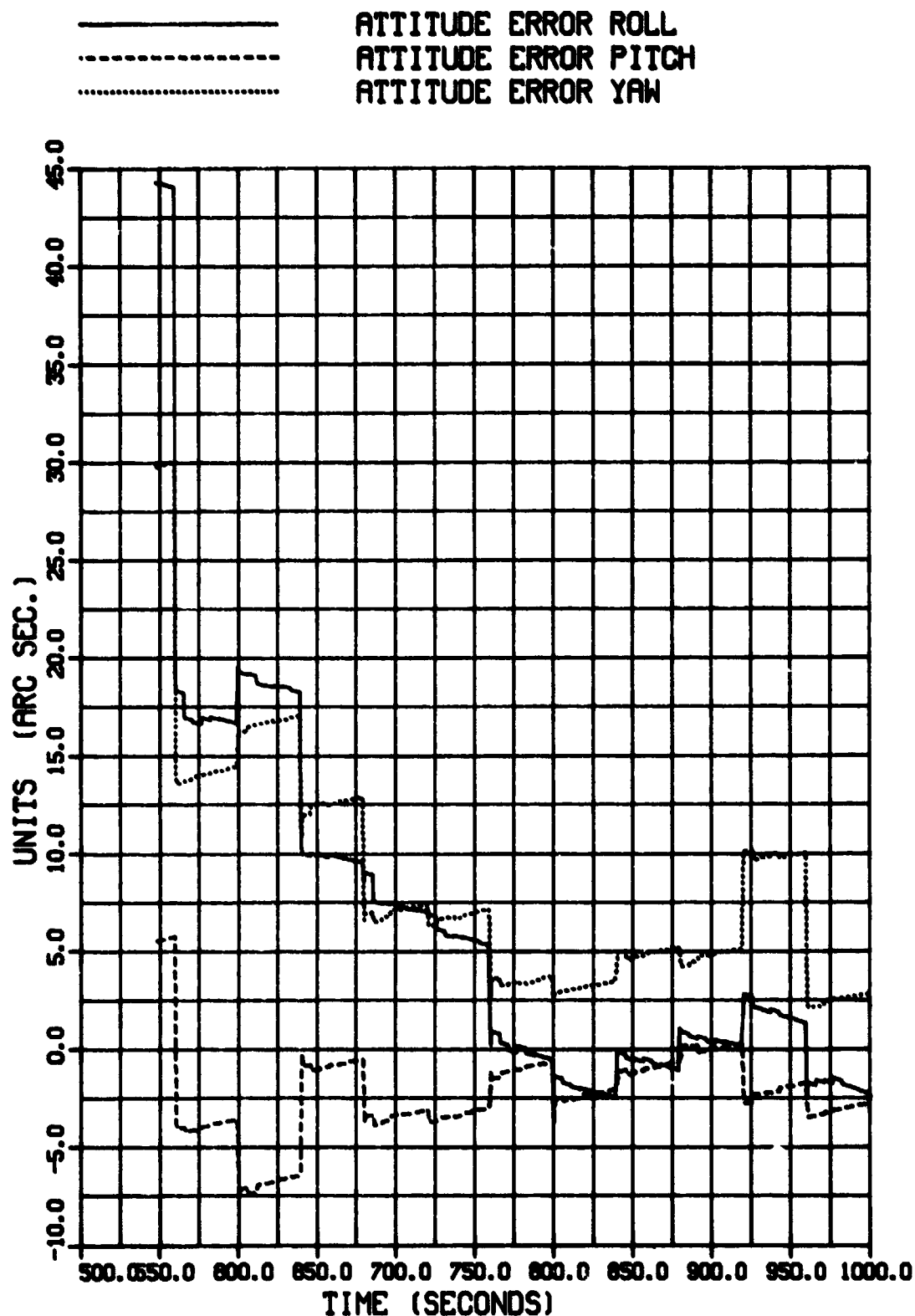


C-2

CASE4-DATA GCP-40SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



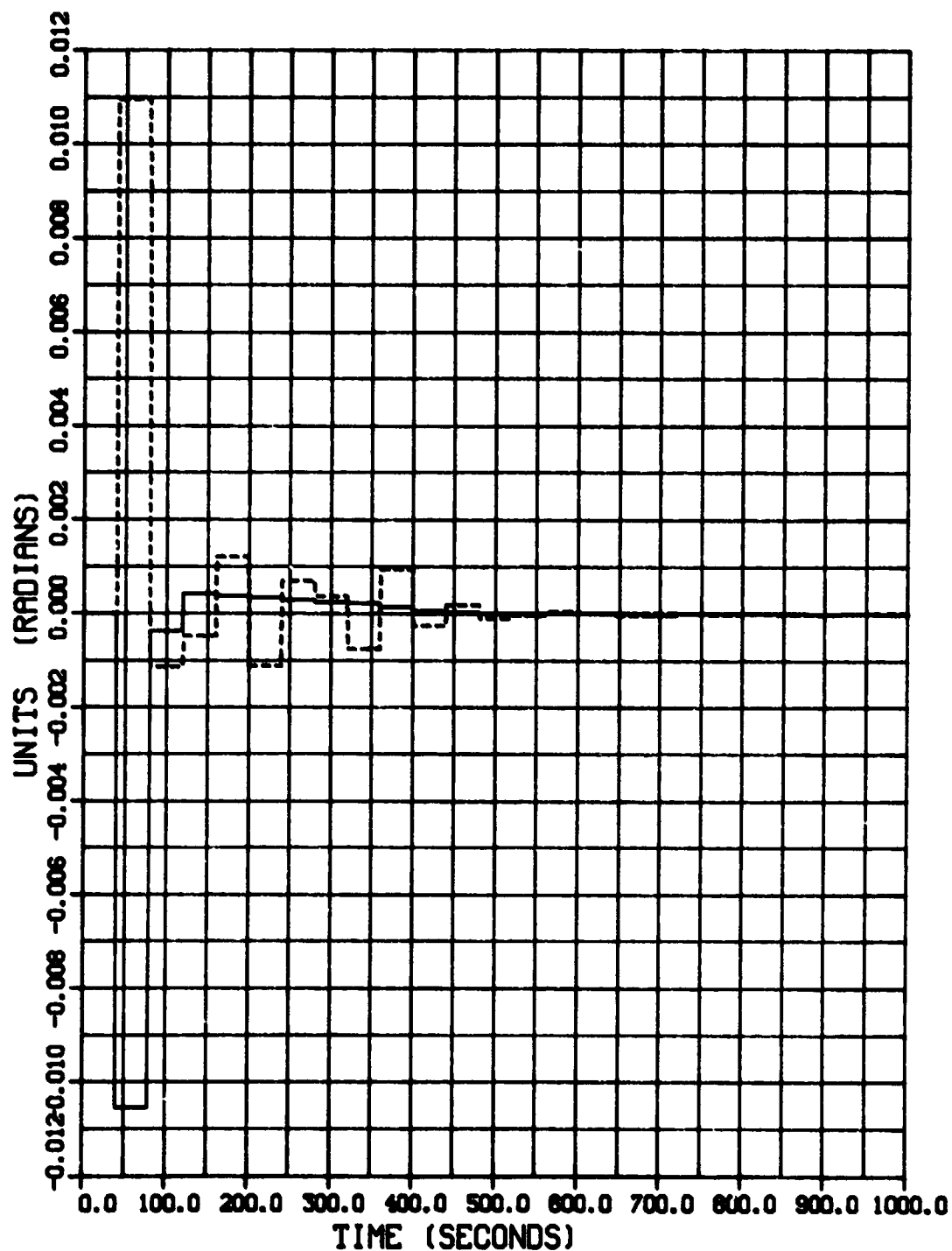
CASE4-DATA GCP-40SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



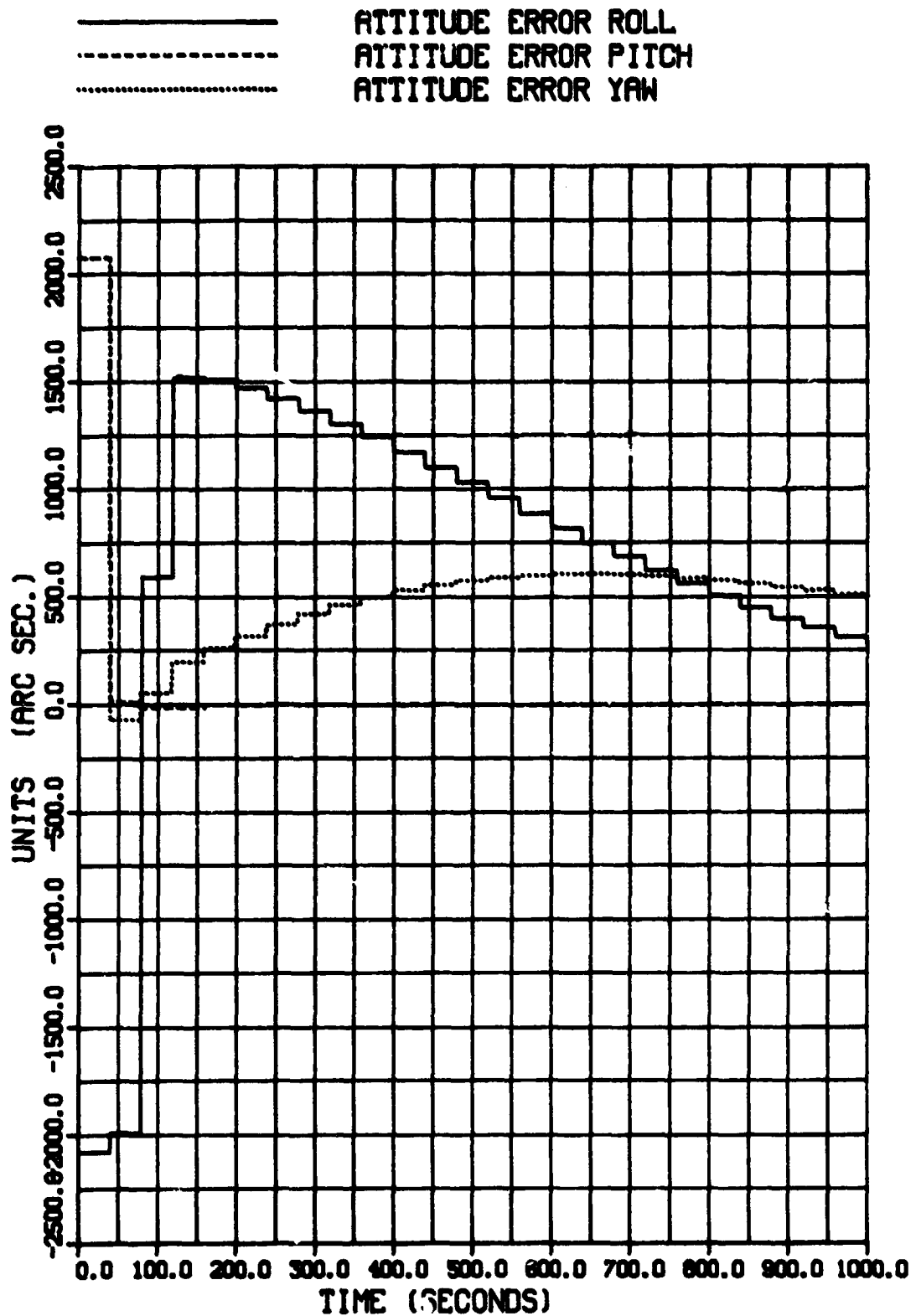
CASE4-DATA GCP-40SEC GPS-5SEC

*** INERTIAL COORDINATE FRAME ***

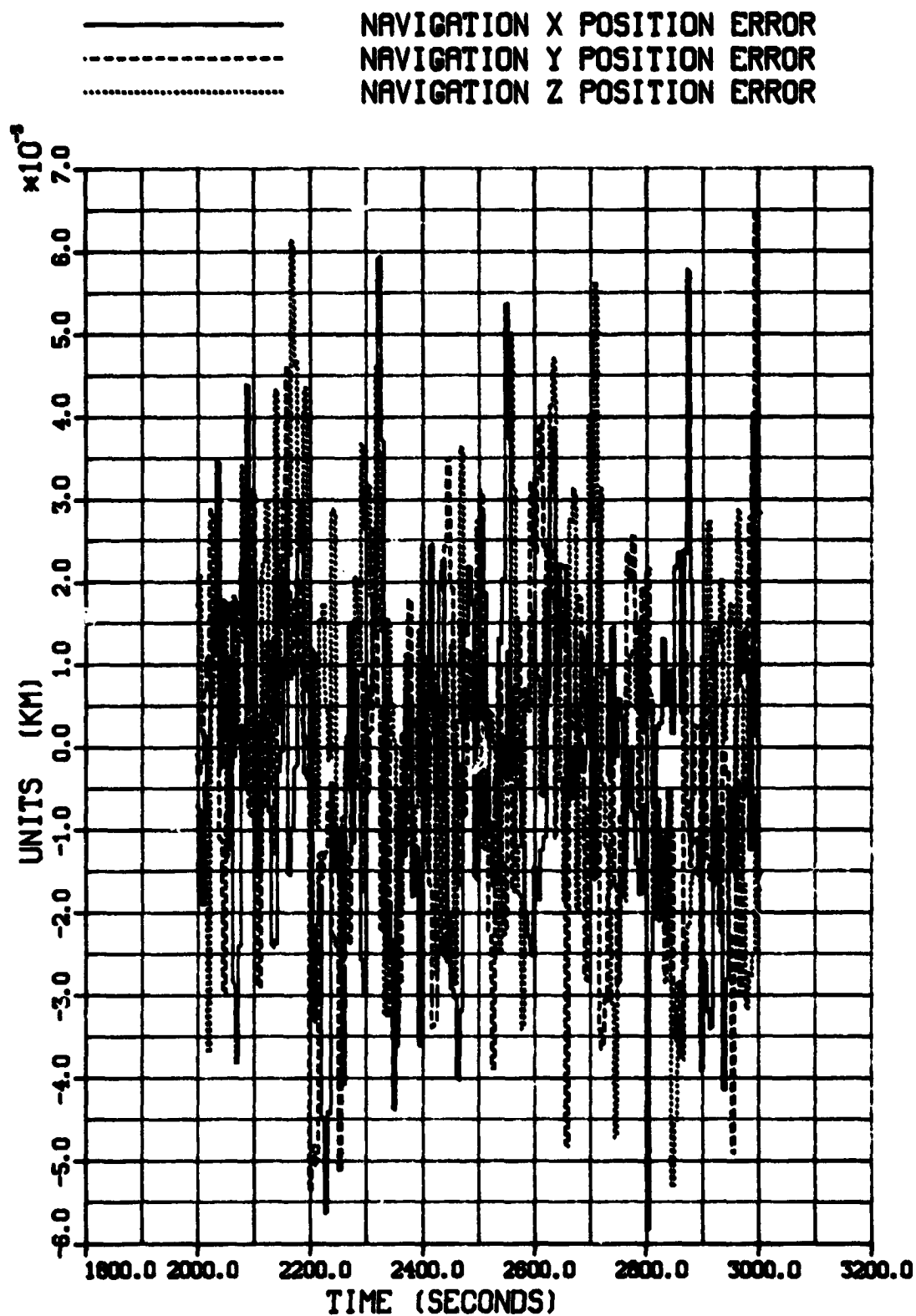
———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL



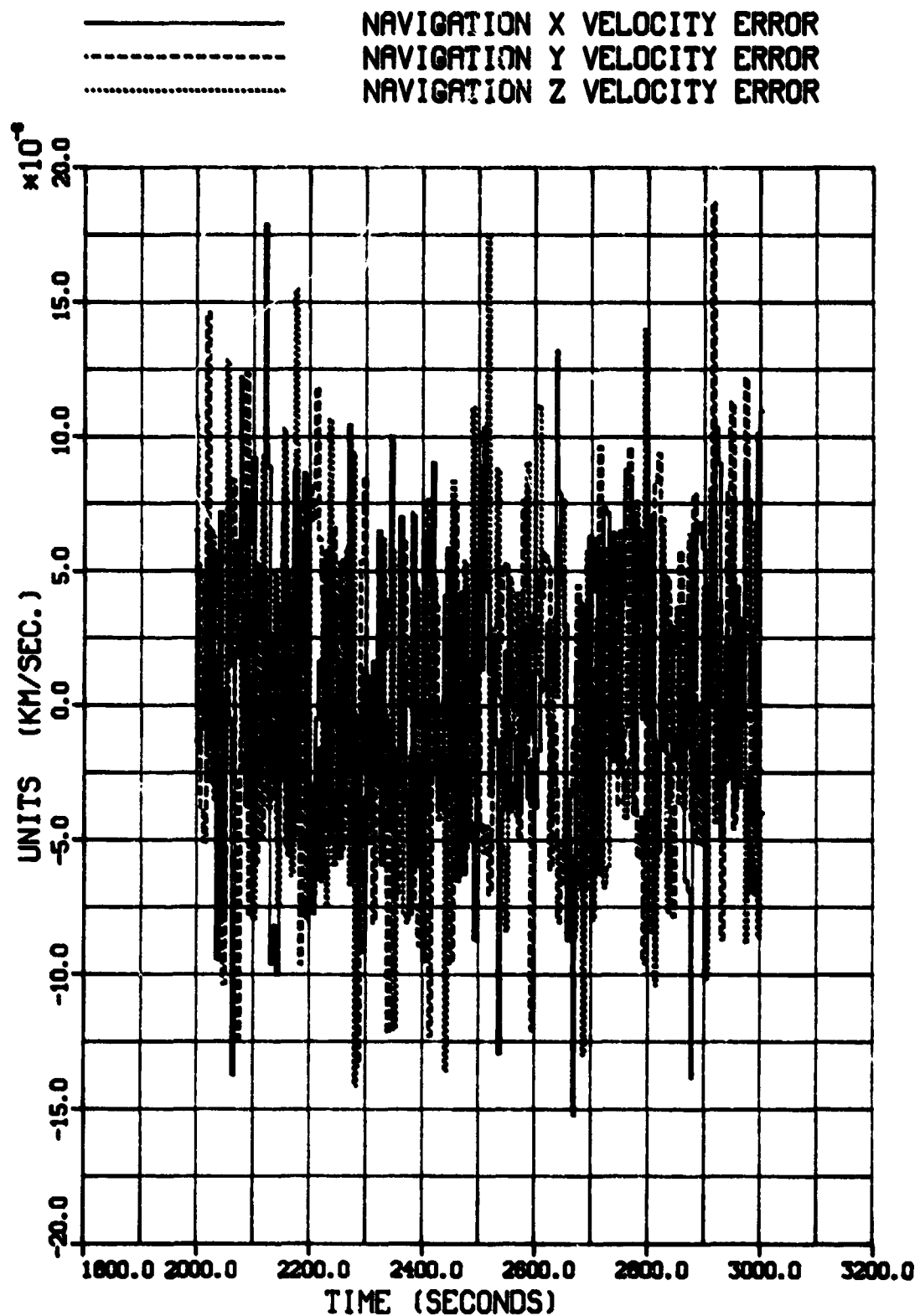
CASE4-DATA GCP-40SEC GPS-5SEC LA - FIXED
 *** INERTIAL COORDINATE FRAME ***



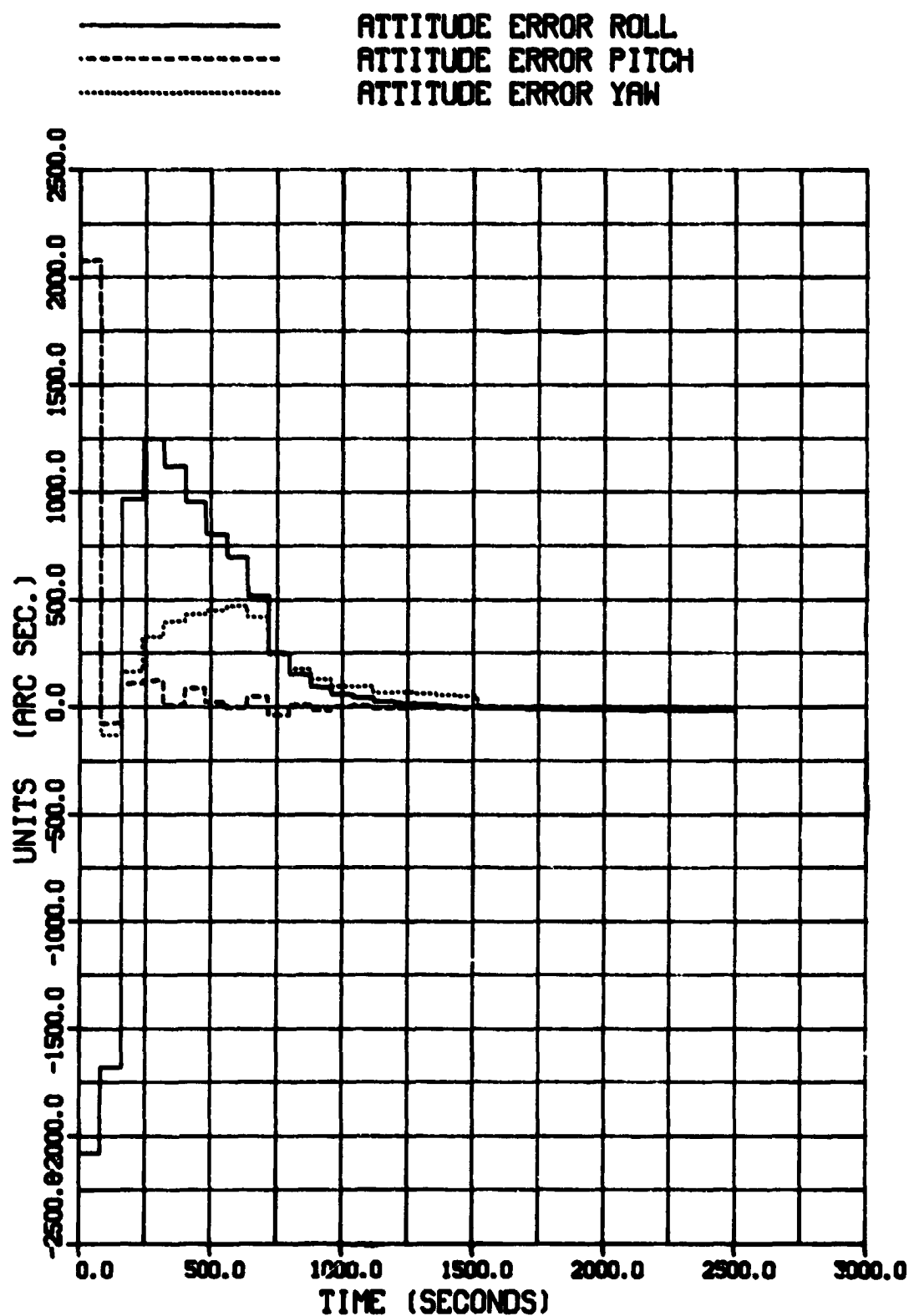
CASE5-DATA GCP-80SEC GPS-5SEC LA-RANDOM
*** INERTIAL COORDINATE FRAME ***



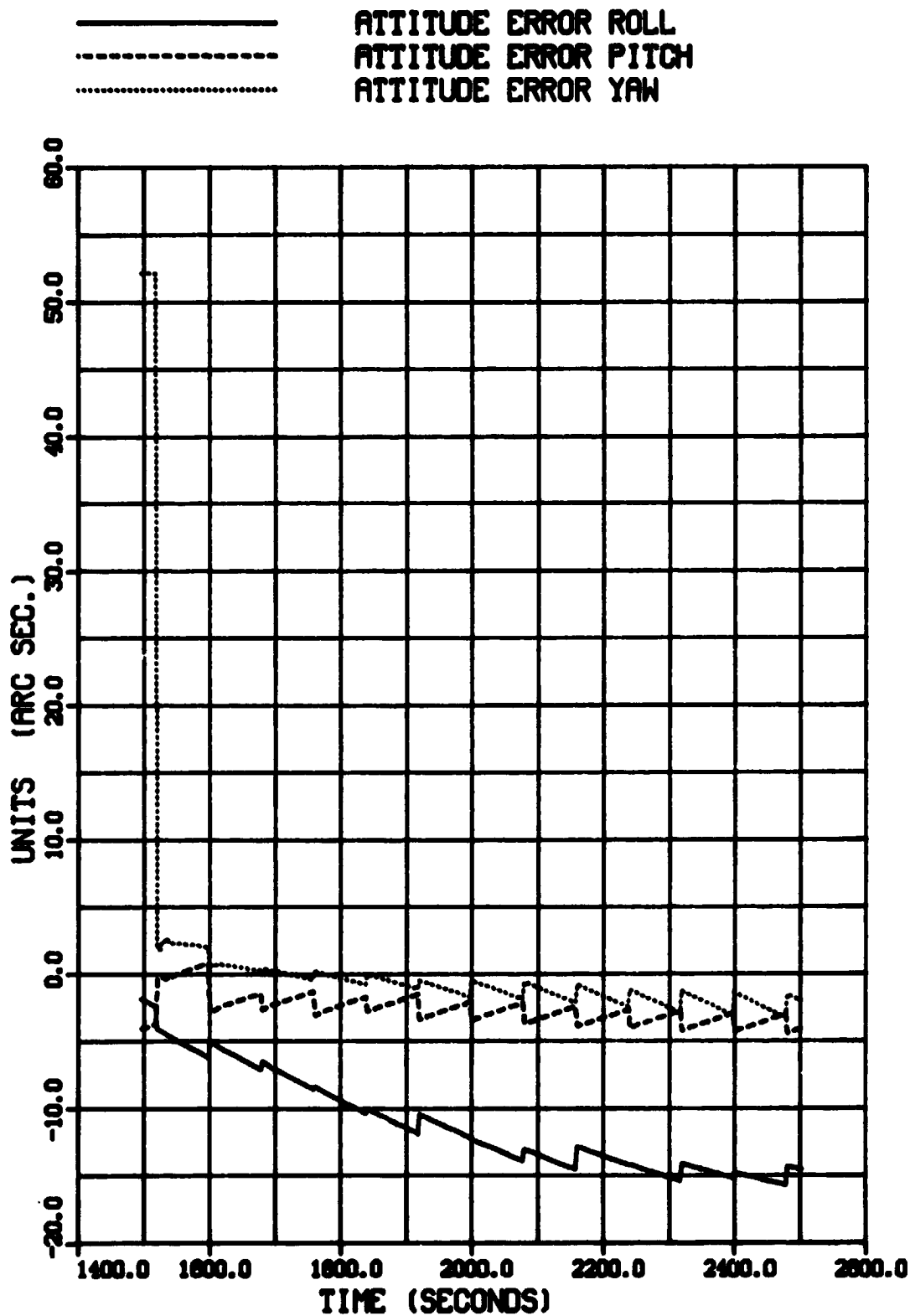
CASE5-DATA GCP-80SEC GPS-5SEC LA-RANLDM
 *** INERTIAL COORDINATE FRAME ***



CASE5-DATA GCP-80SEC GPS-5SEC LA-RANDOM
*** INERTIAL COORDINATE FRAME ***

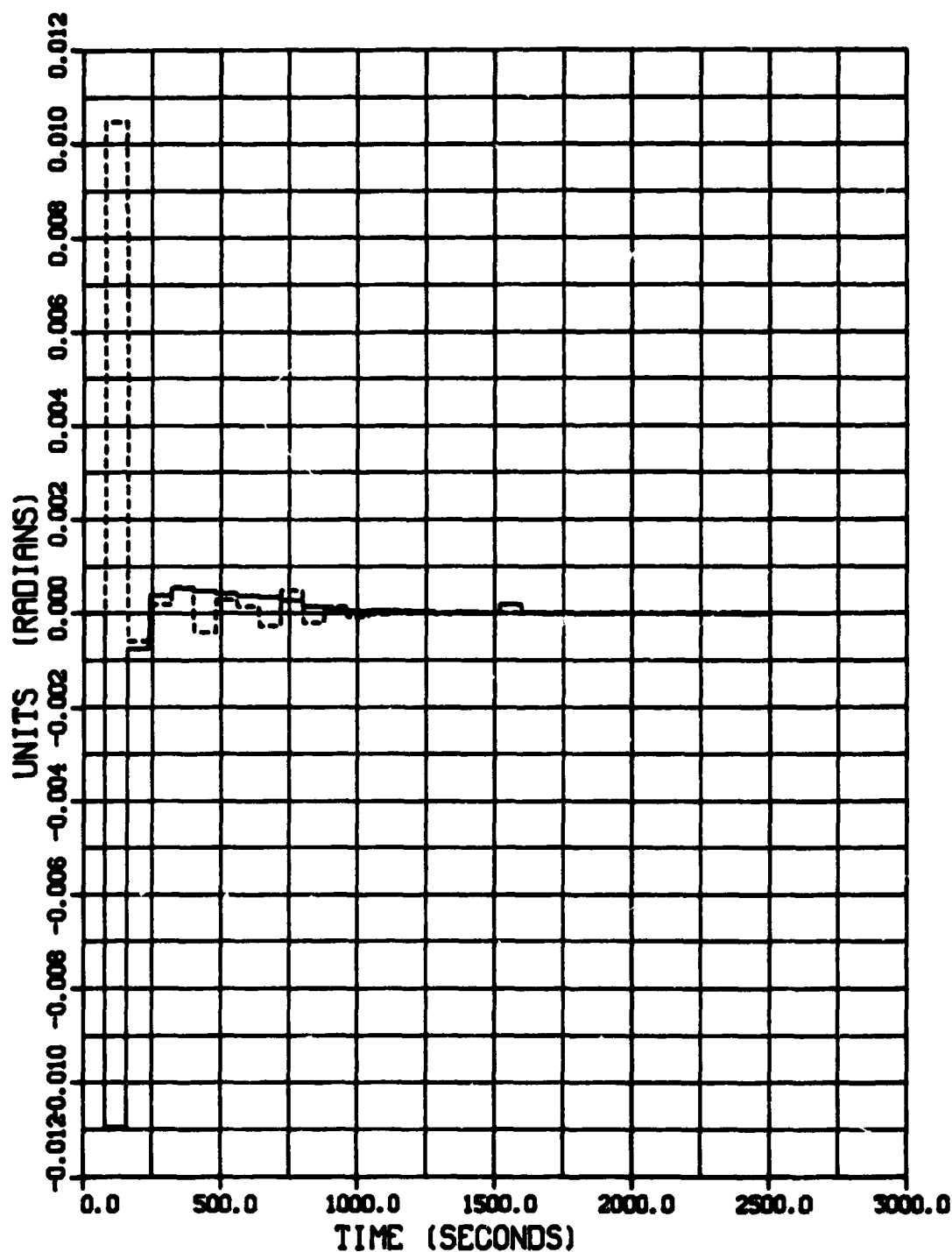


CASE5-DATA GCP-80SEC GPS-5SEC LA-RANDOM
*** INERTIAL COORDINATE FRAME ***

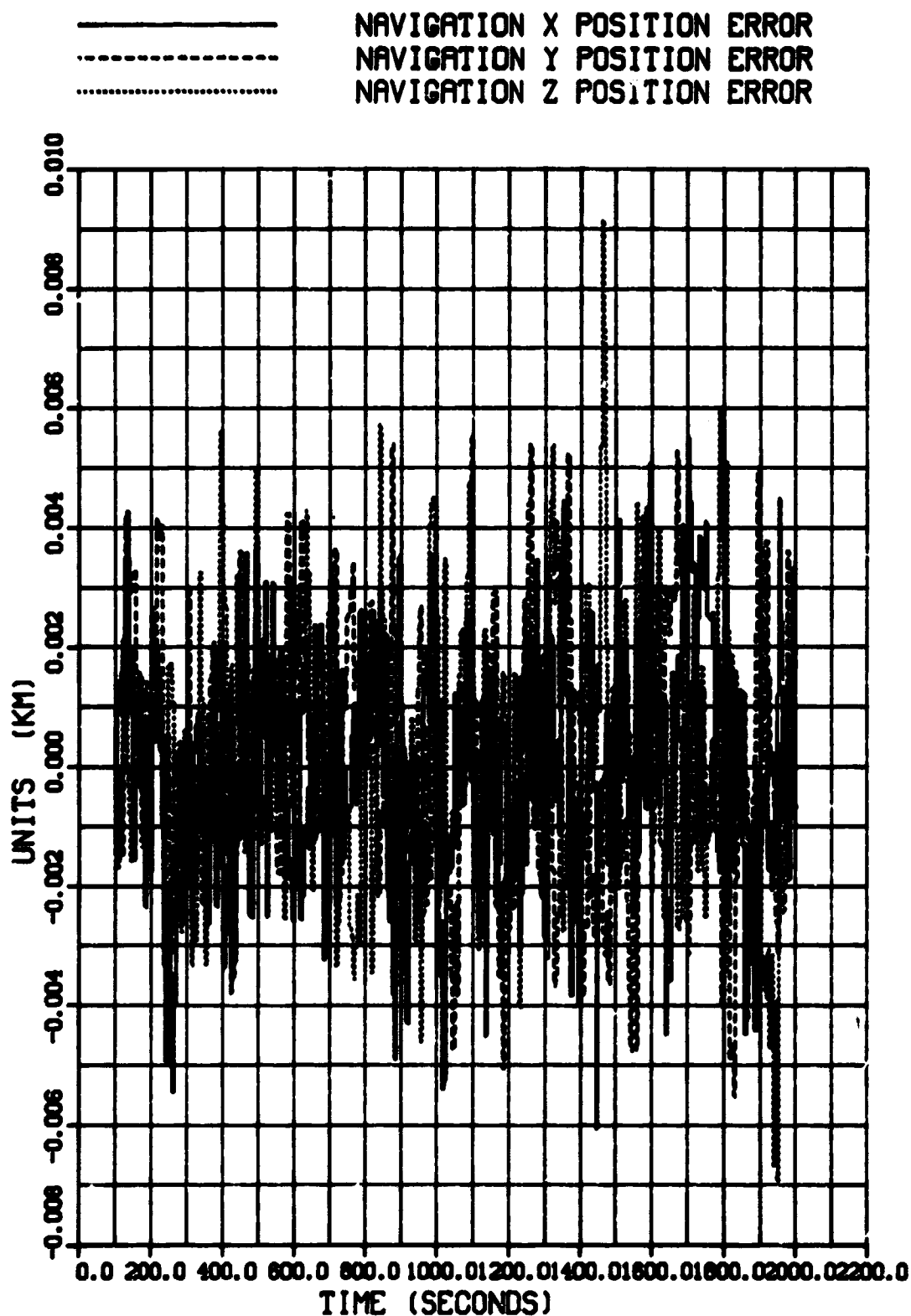


CASE5-DATA GCP-80SEC GPS-5SEC LA-RANDOM
*** INERTIAL COORDINATE FRAME ***

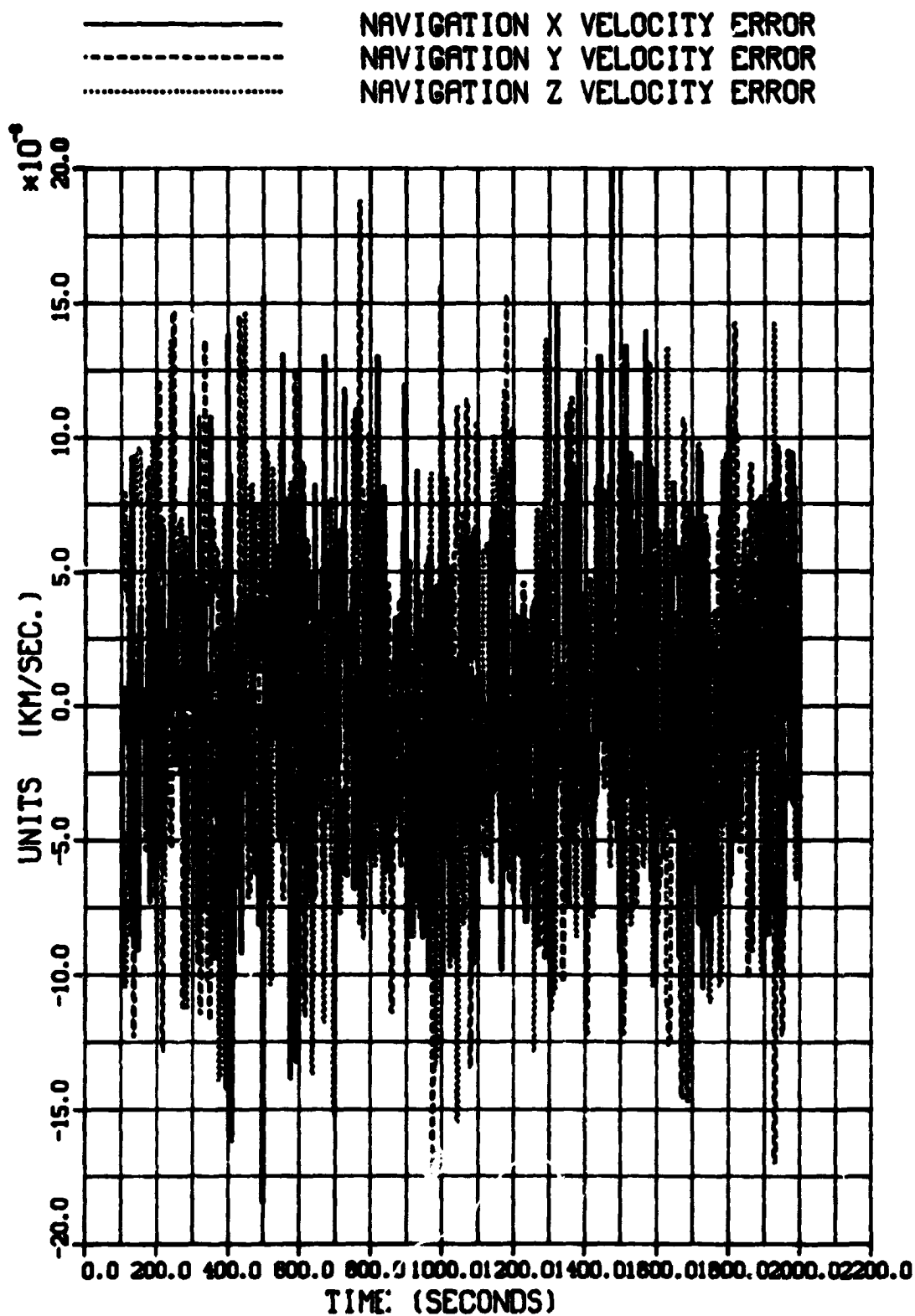
———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL



CASE6-DATA GCP-160SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

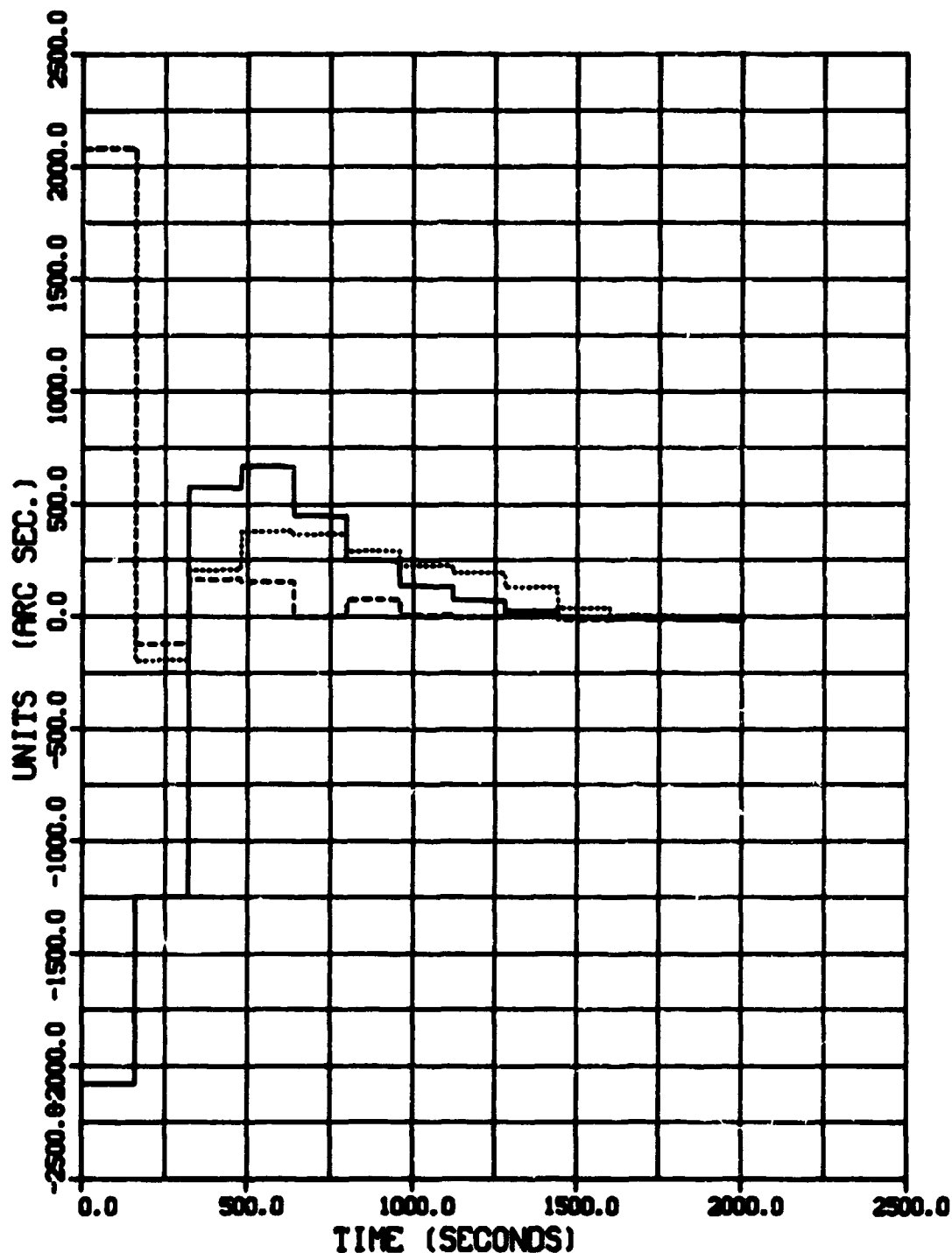


CASE6-DATA GCP-160SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***



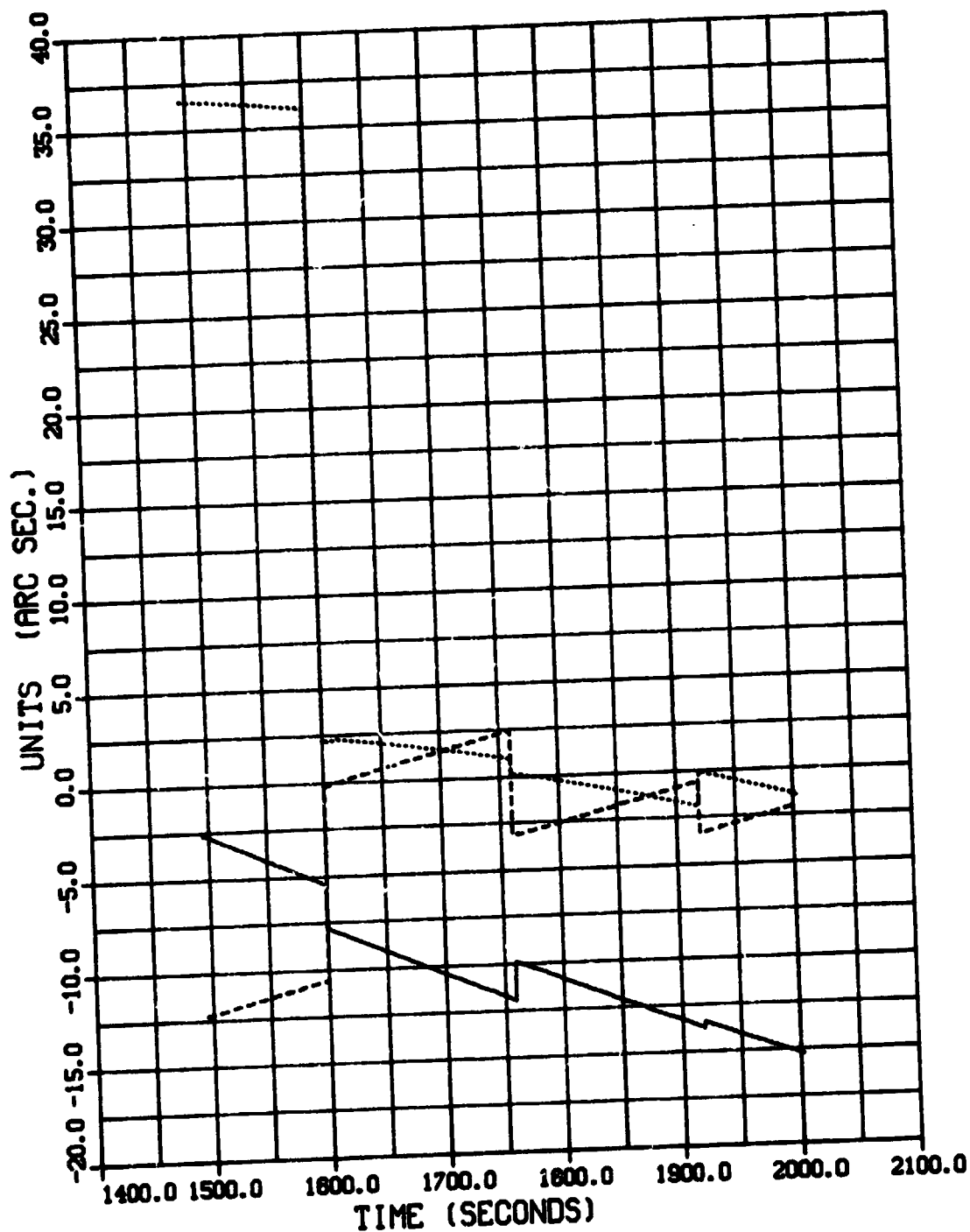
CASE6-DATA GCP-160SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***

———— ATTITUDE ERROR ROLL
 ATTITUDE ERROR PITCH
 ATTITUDE ERROR YAW



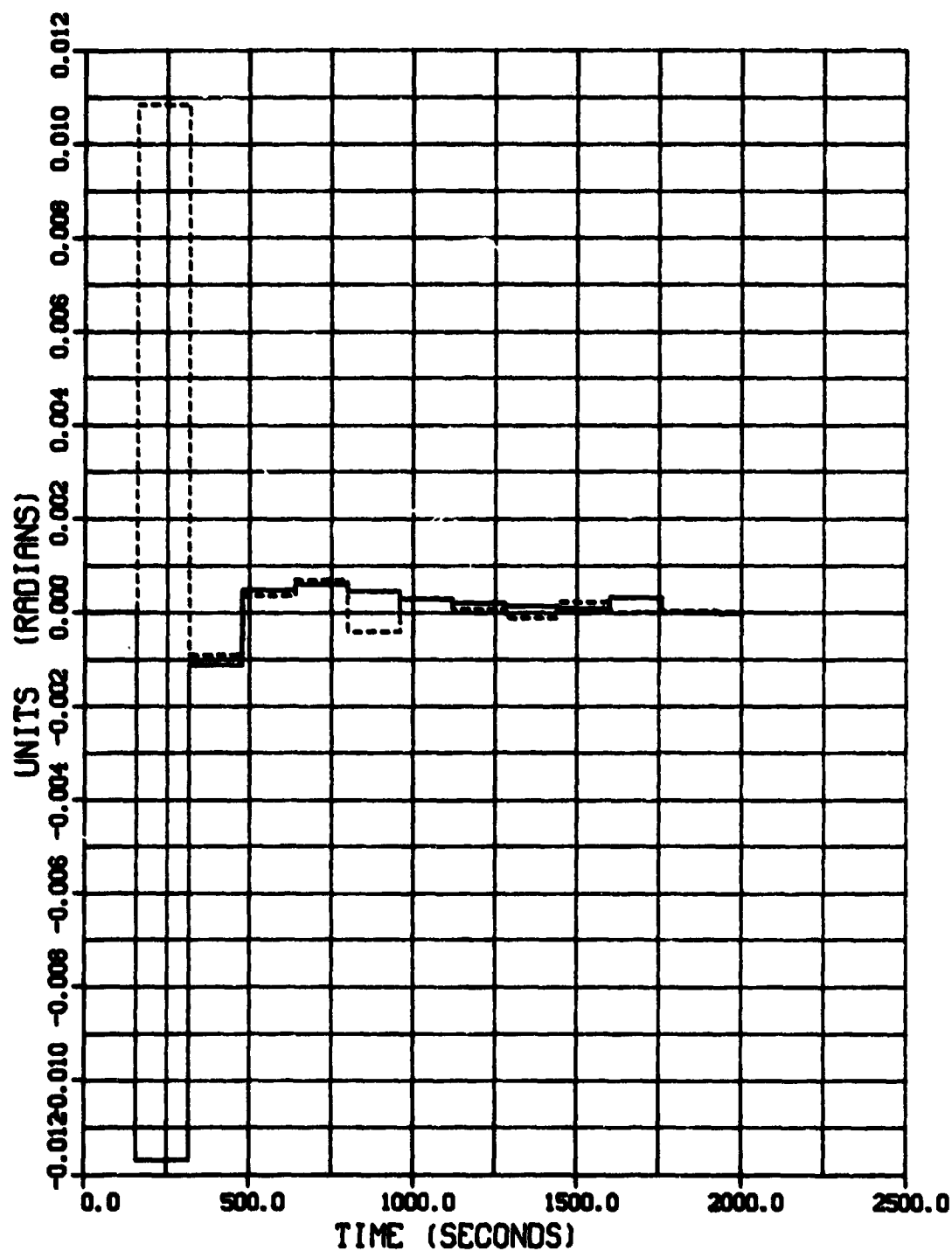
CASE6-DATA GCP-160SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***

_____ ATTITUDE ERROR ROLL
 - - - - - ATTITUDE ERROR PITCH
 ATTITUDE ERROR YAW

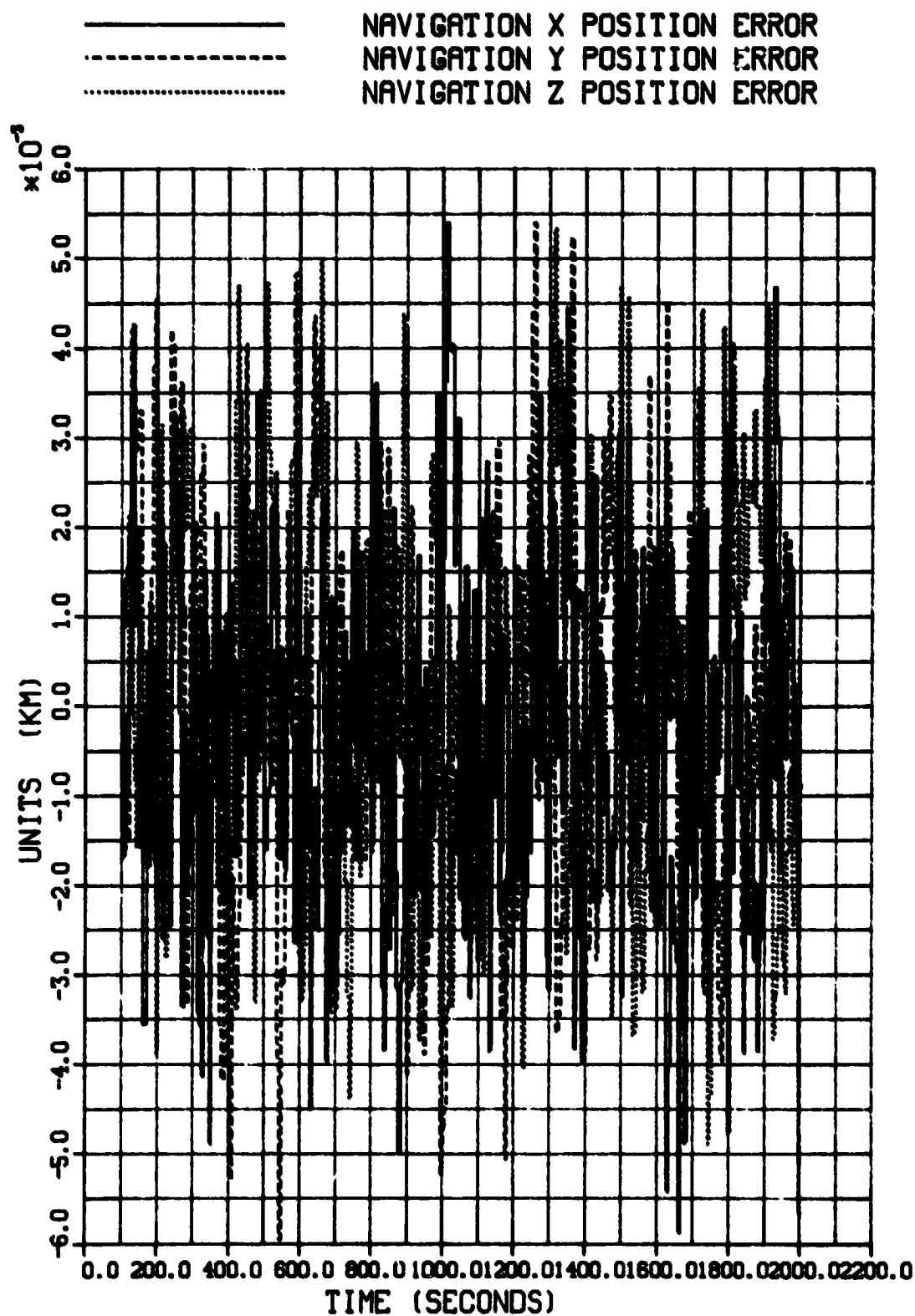


CASE6-DATA GCP-160SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

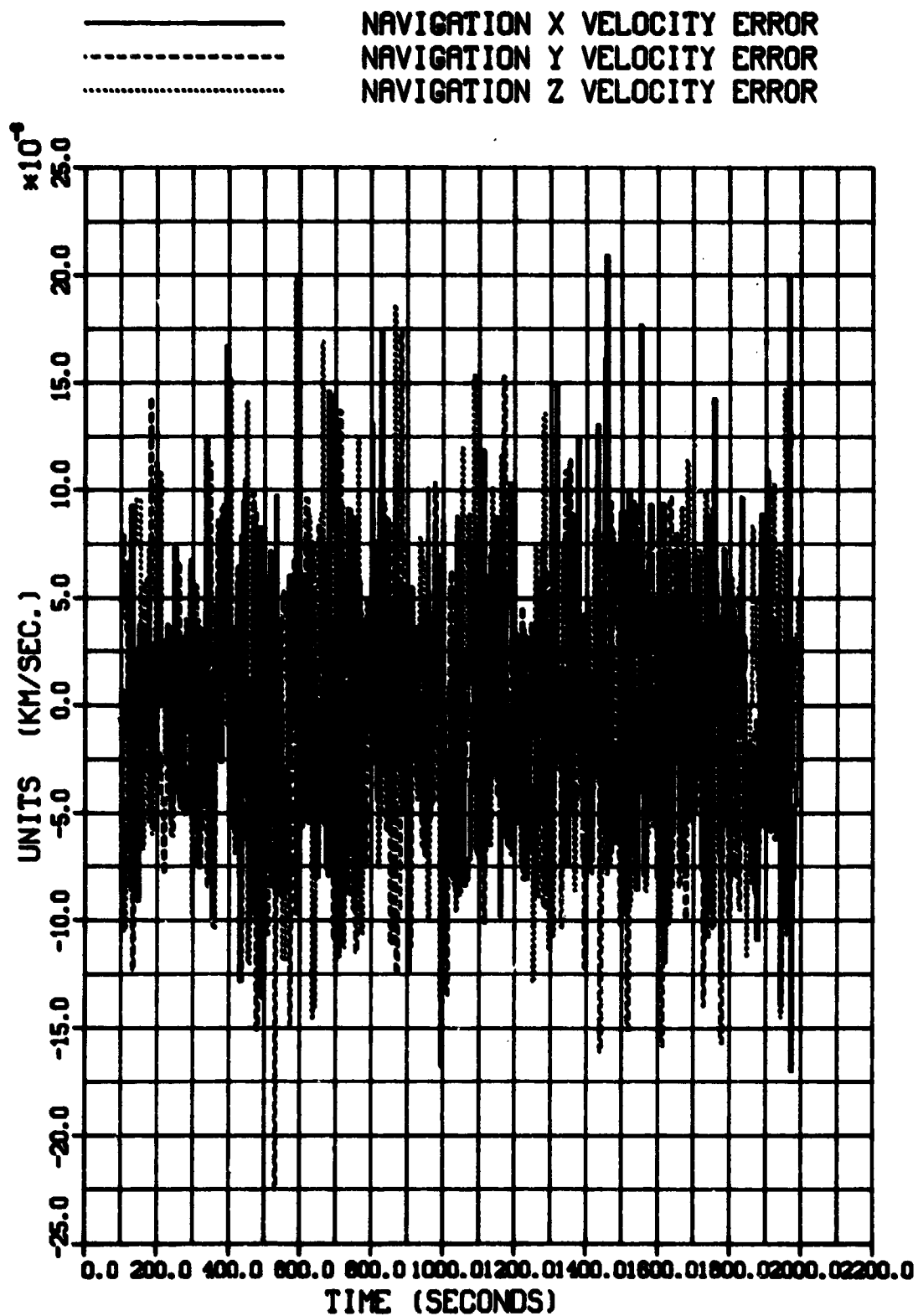
———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL



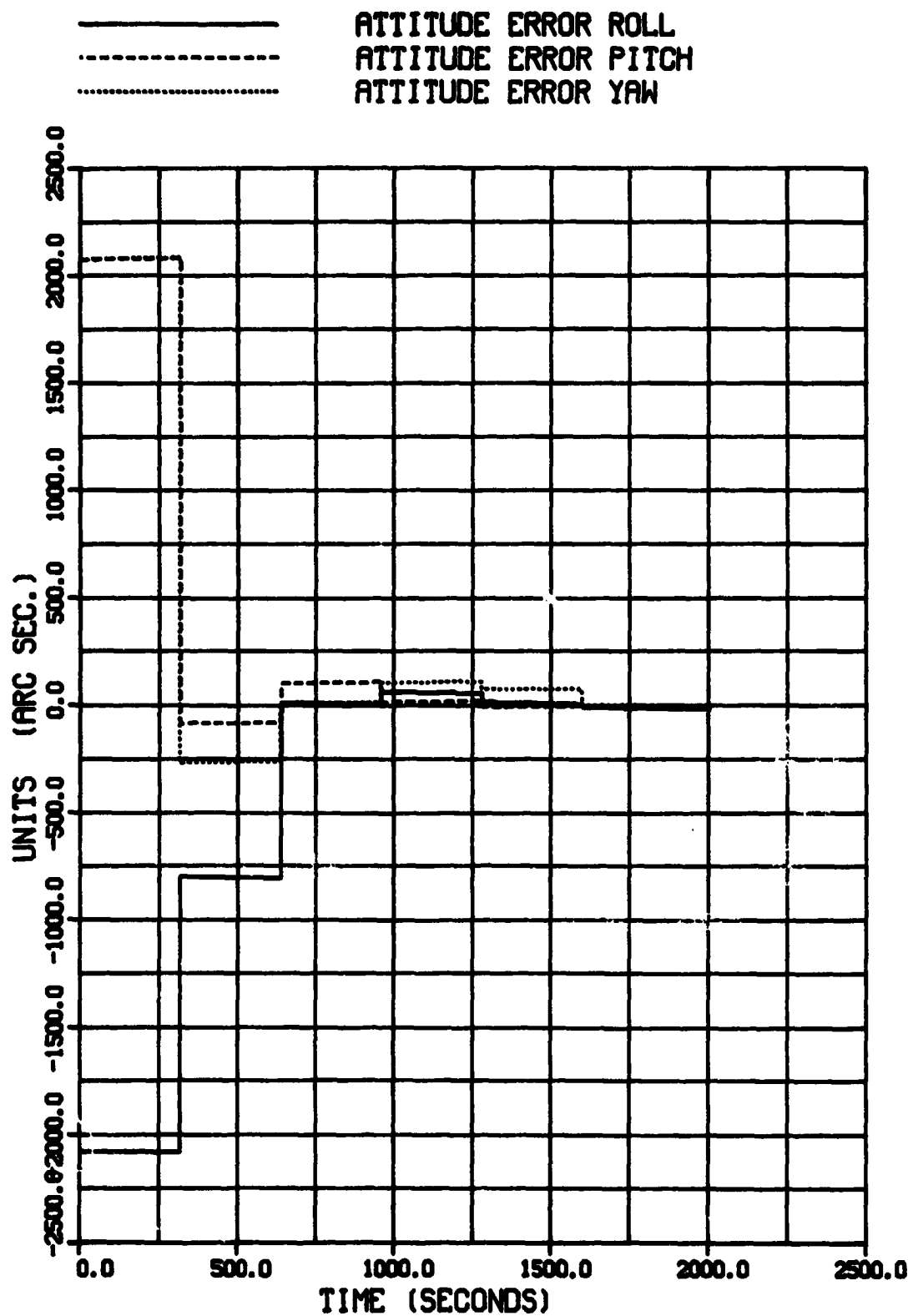
CASE7-DATA GCP-320SEC GPS-5SEC
 *** INERTIAL COORDINATE FRAME ***



CASE7-DATA GCP-320SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

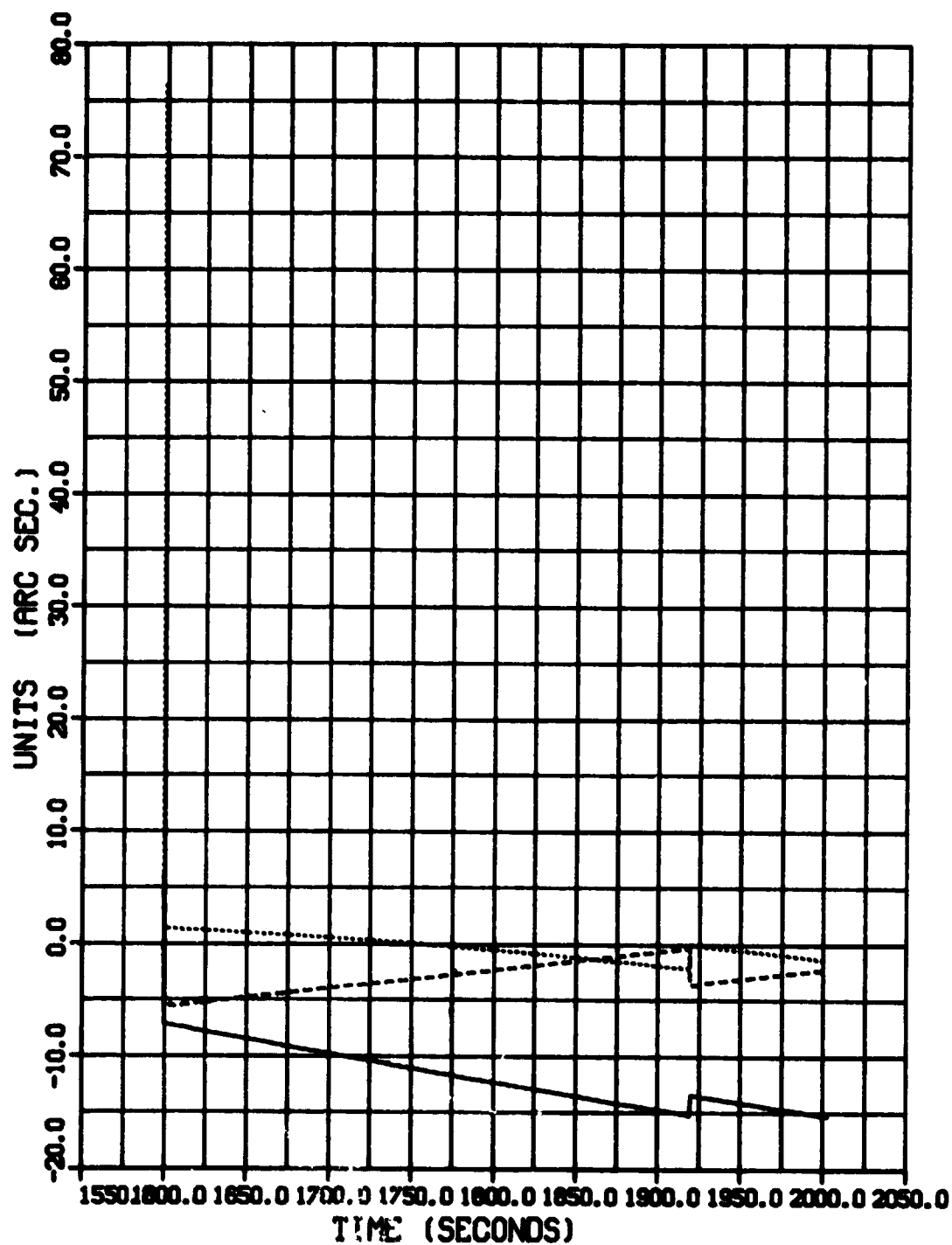


CASE7-DATA GCP-320SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***



CASE7-DATA GCP-320SEC GPS-5SEC
*** INERTIAL COORDINATE FRAME ***

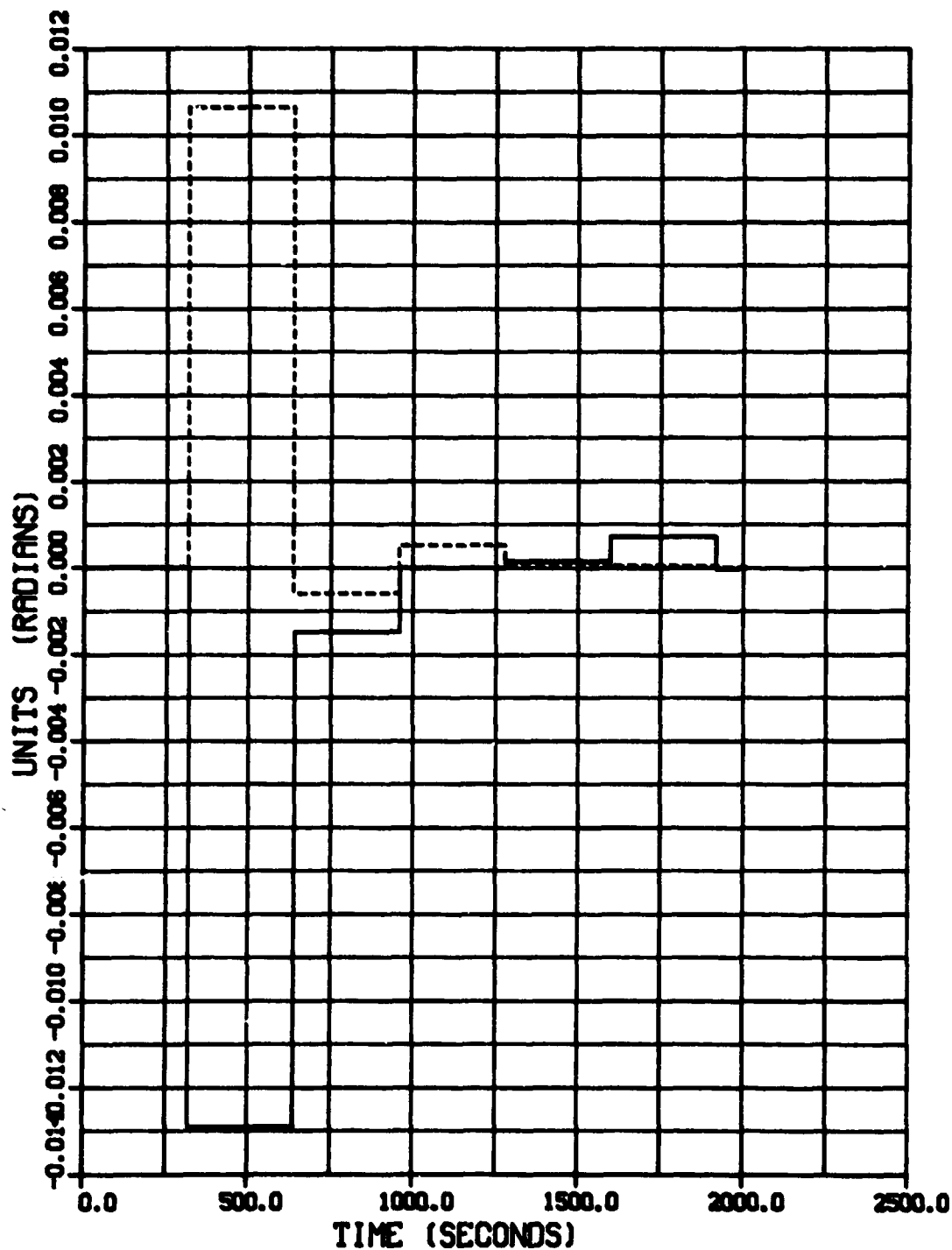
———— ATTITUDE ERROR ROLL
----- ATTITUDE ERROR PITCH
..... ATTITUDE ERROR YAW



CASE7-DATA GCP-320SEC GPS-5SEC

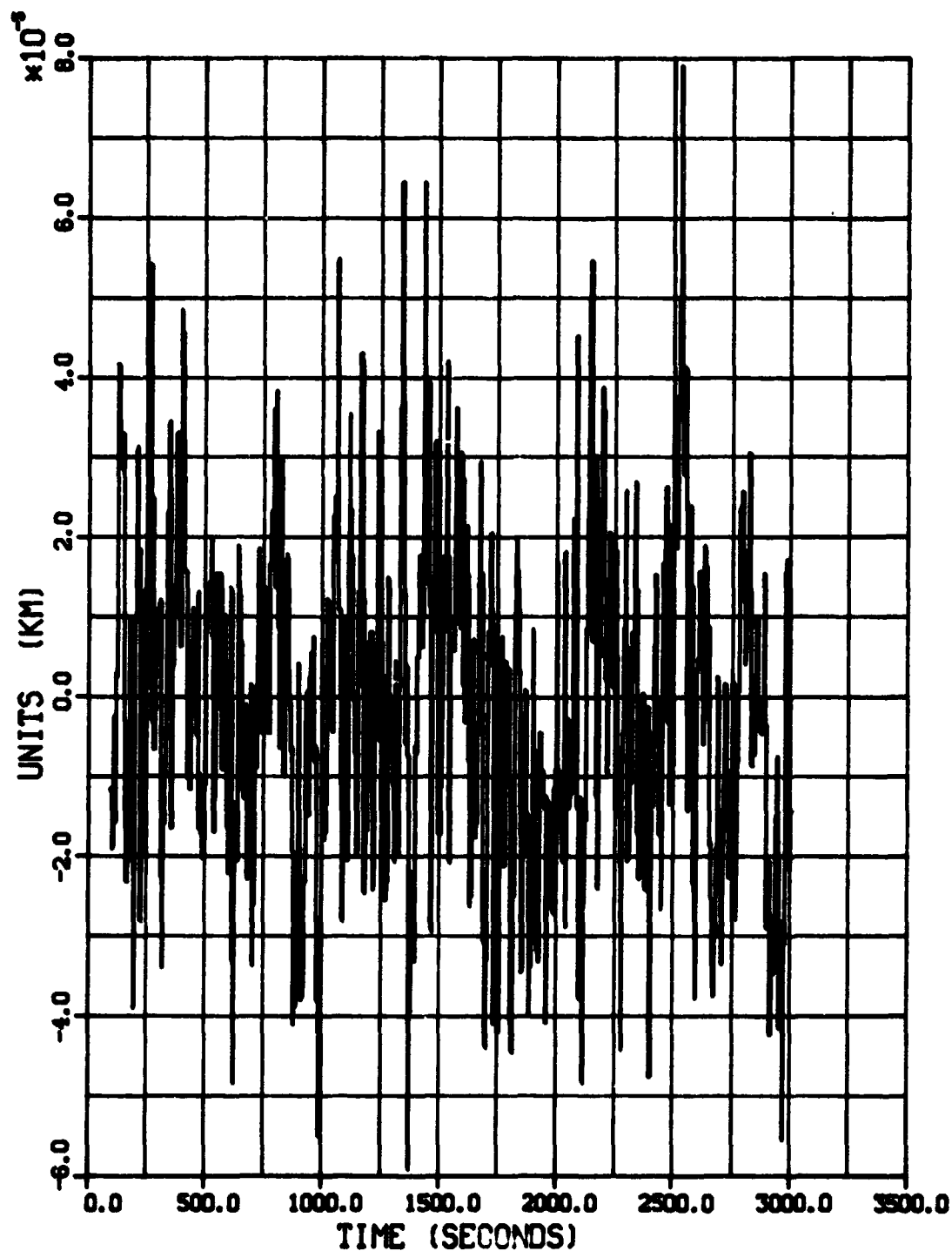
*** INERTIAL COORDINATE FRAME ***

—— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

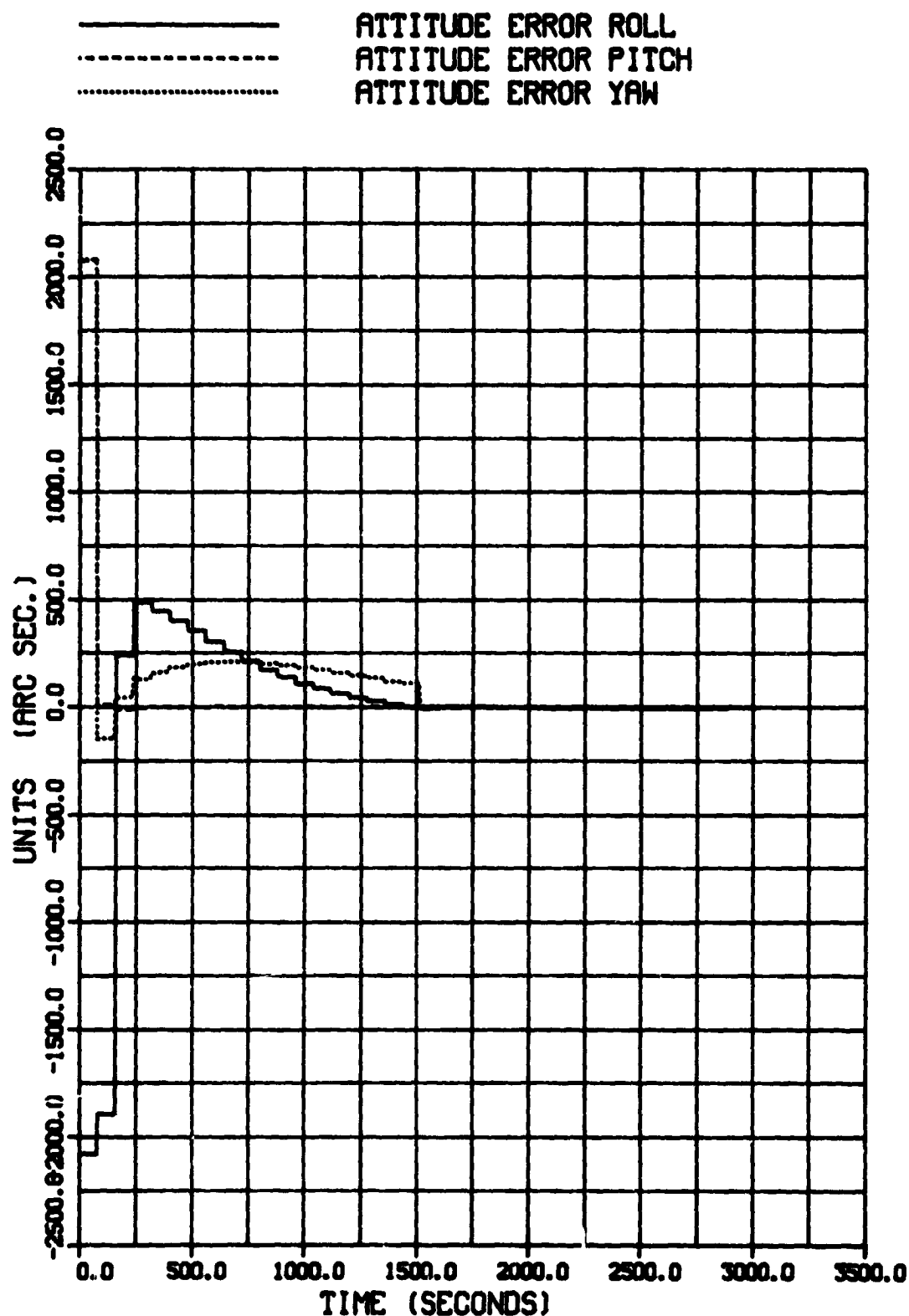


CASE9-DATA GCP-40SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

NAVIGATION X POSITION ERROR

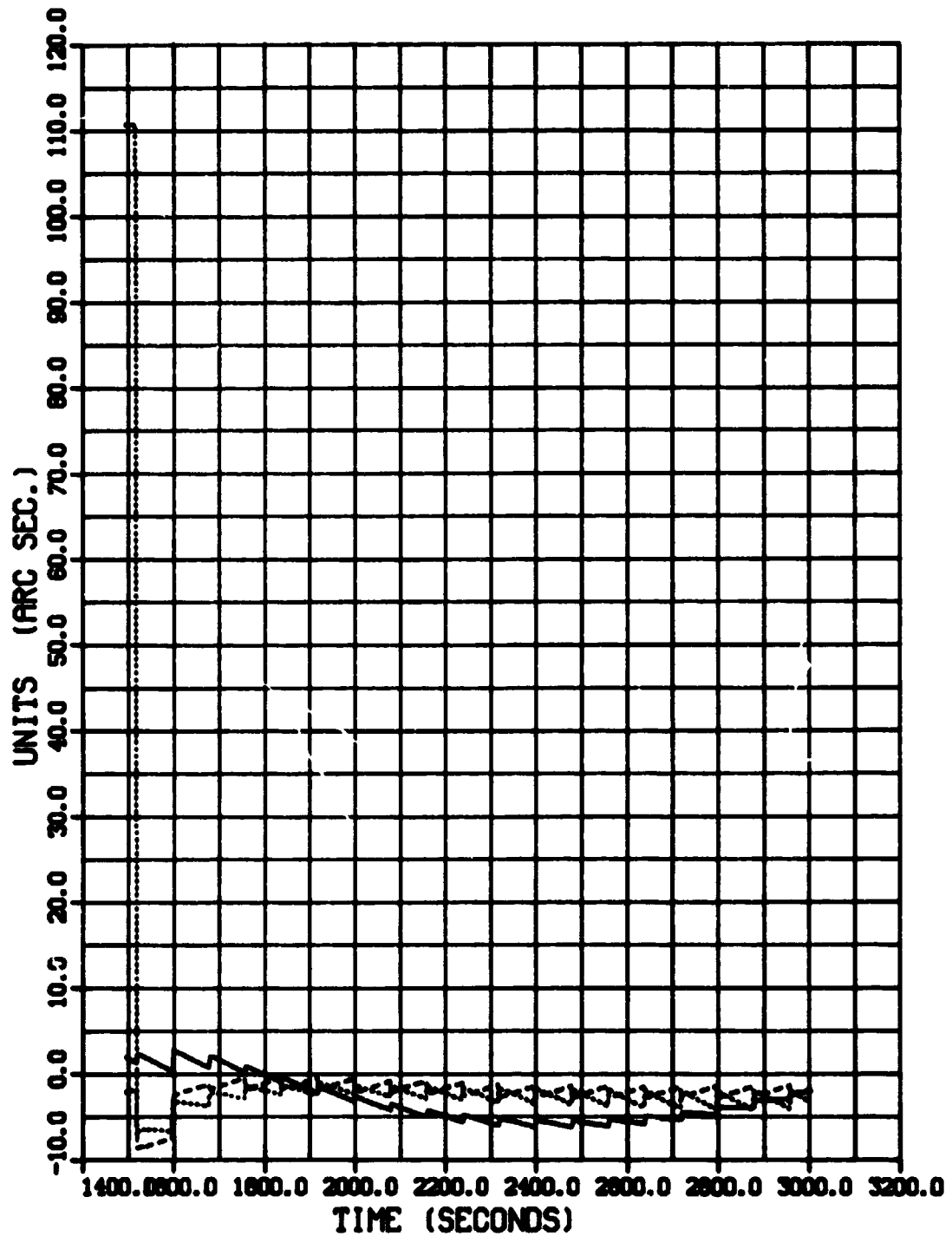


CASE9-DATA GCP-40SEC GPS-5SEC LA-FIXED
 *** INERTIAL COORDINATE FRAME ***



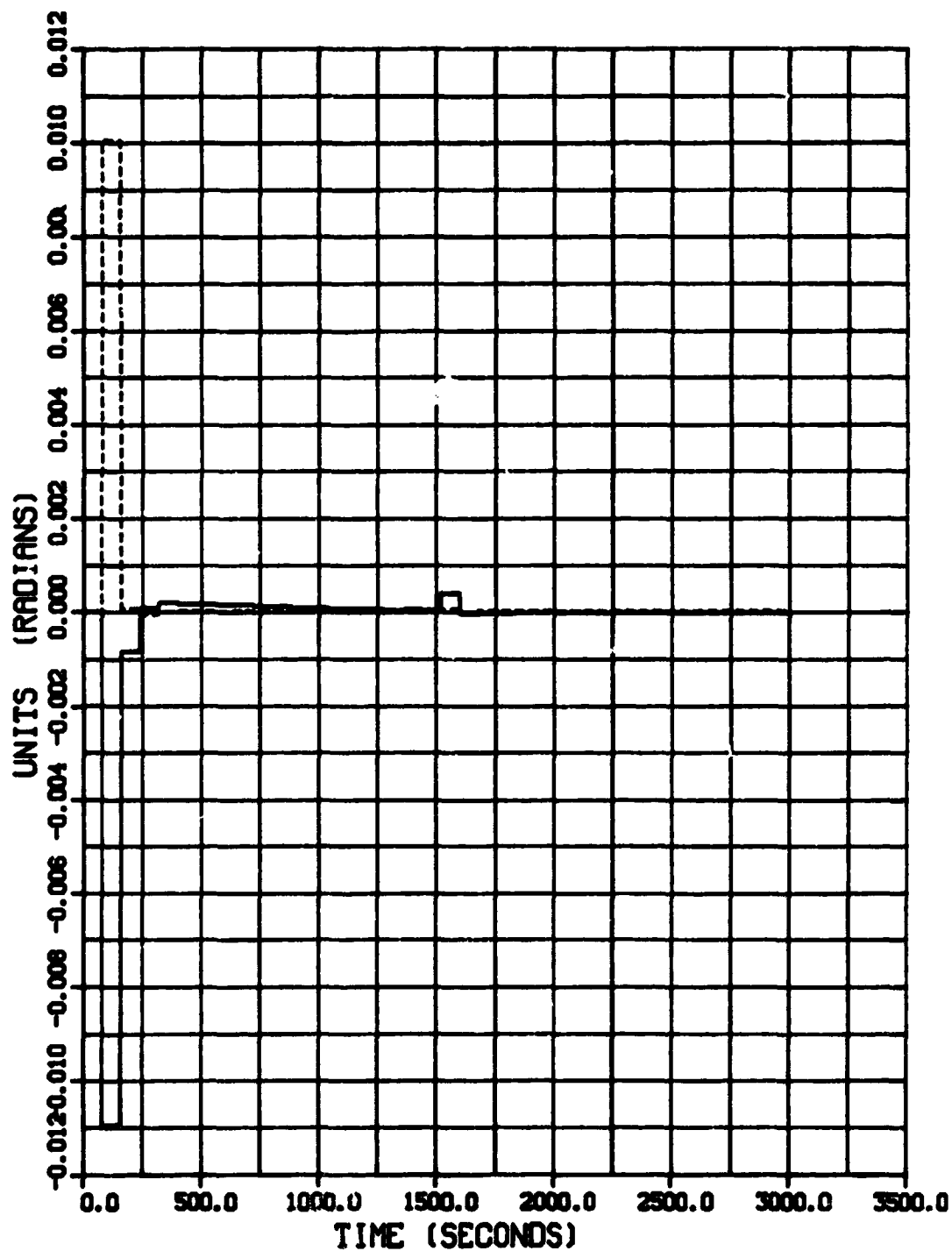
CASE9-DATA GCP-40SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

————— ATTITUDE ERROR ROLL
- - - - - ATTITUDE ERROR PITCH
..... ATTITUDE ERROR YAW



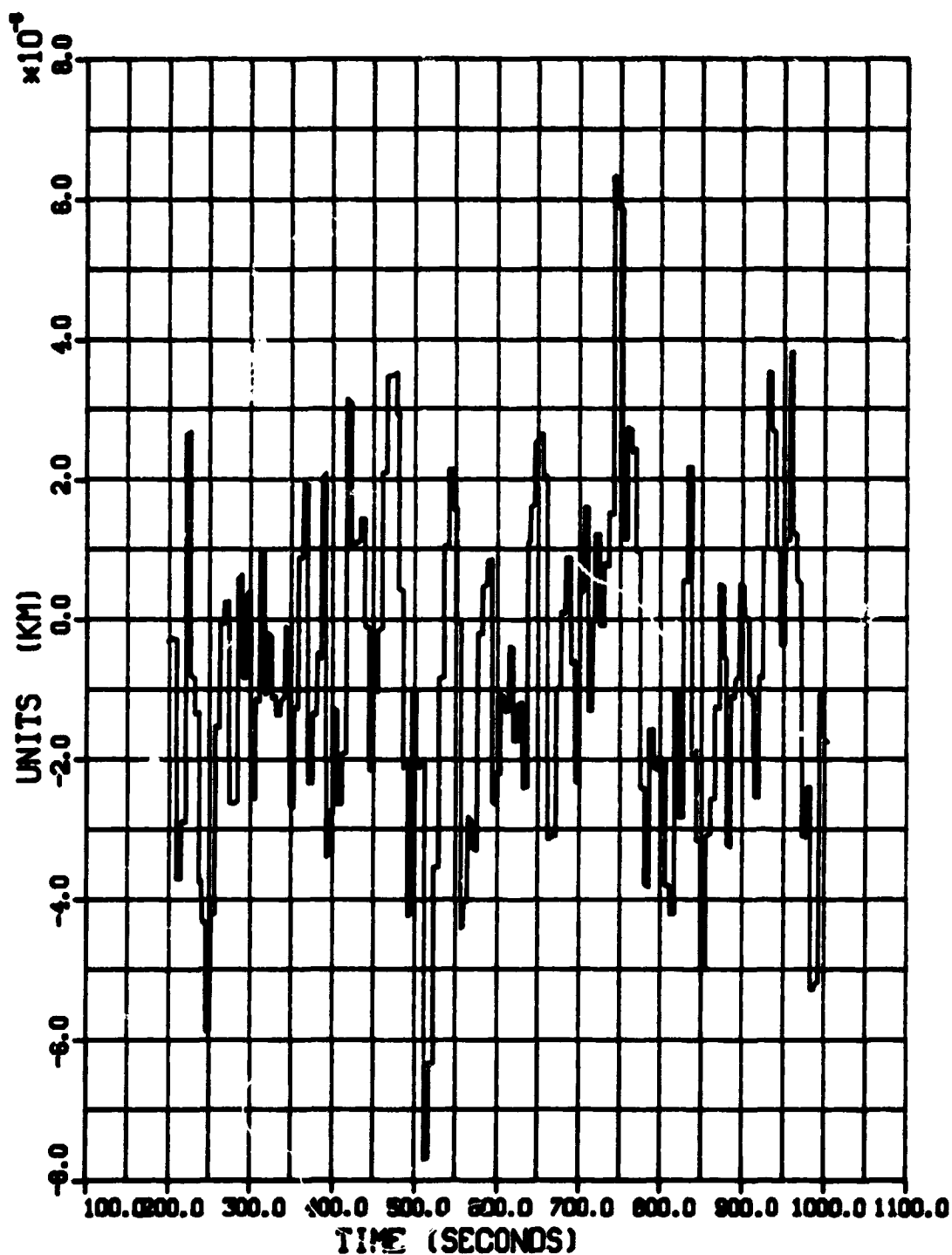
CASE9-DATA GCP-40SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

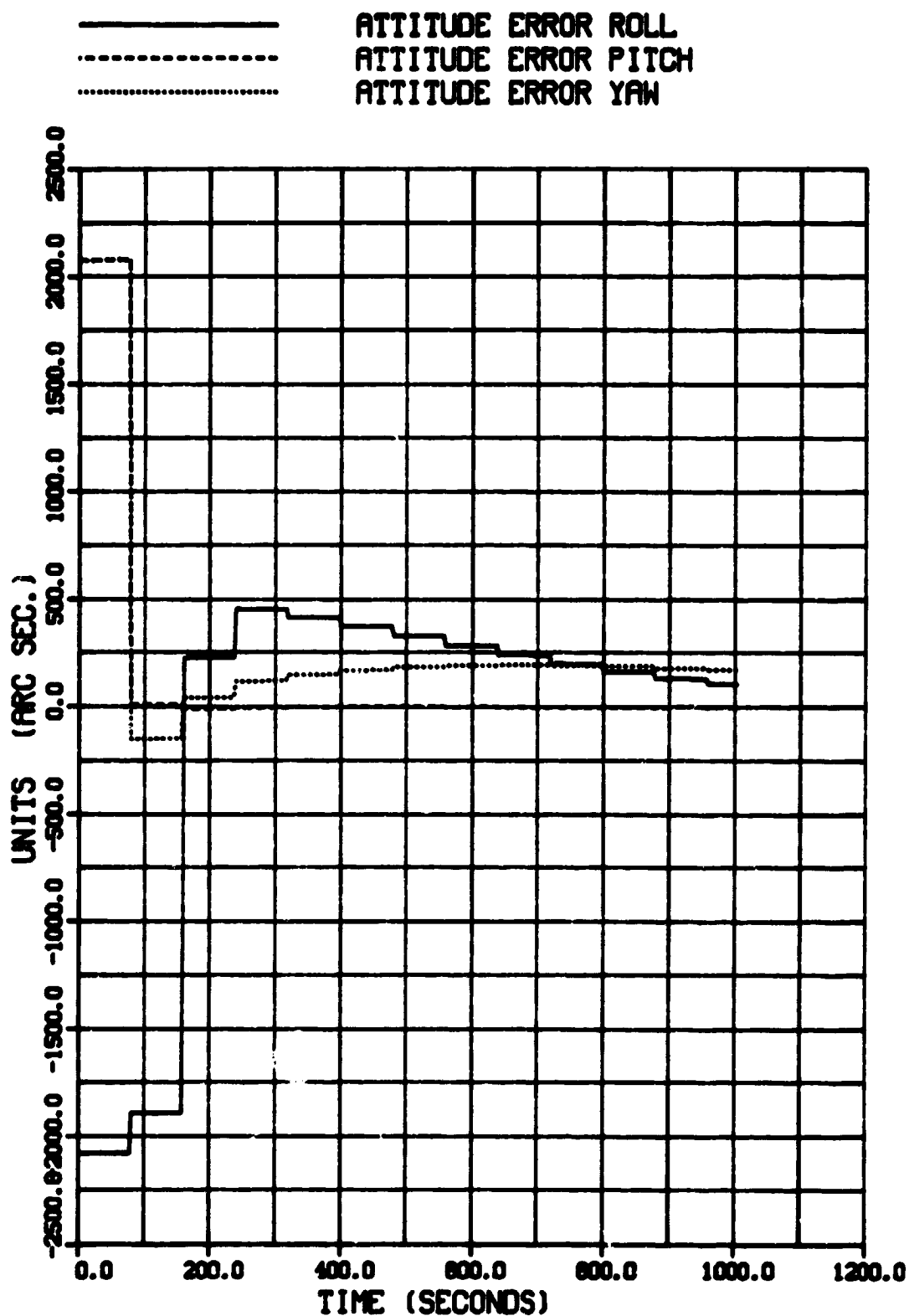


CASE10-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

NAVIGATION X POSITION ERROR

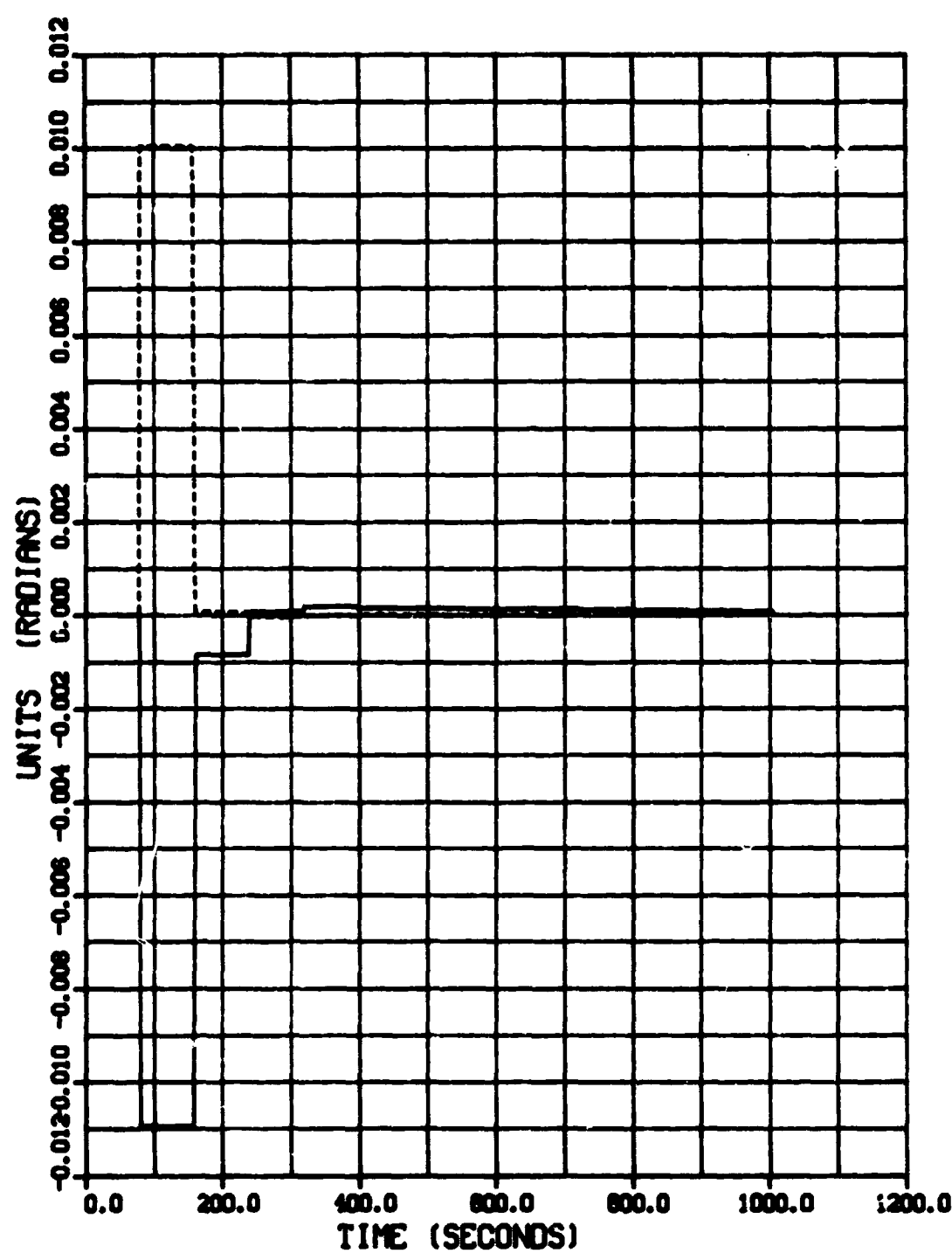


CASE10-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***



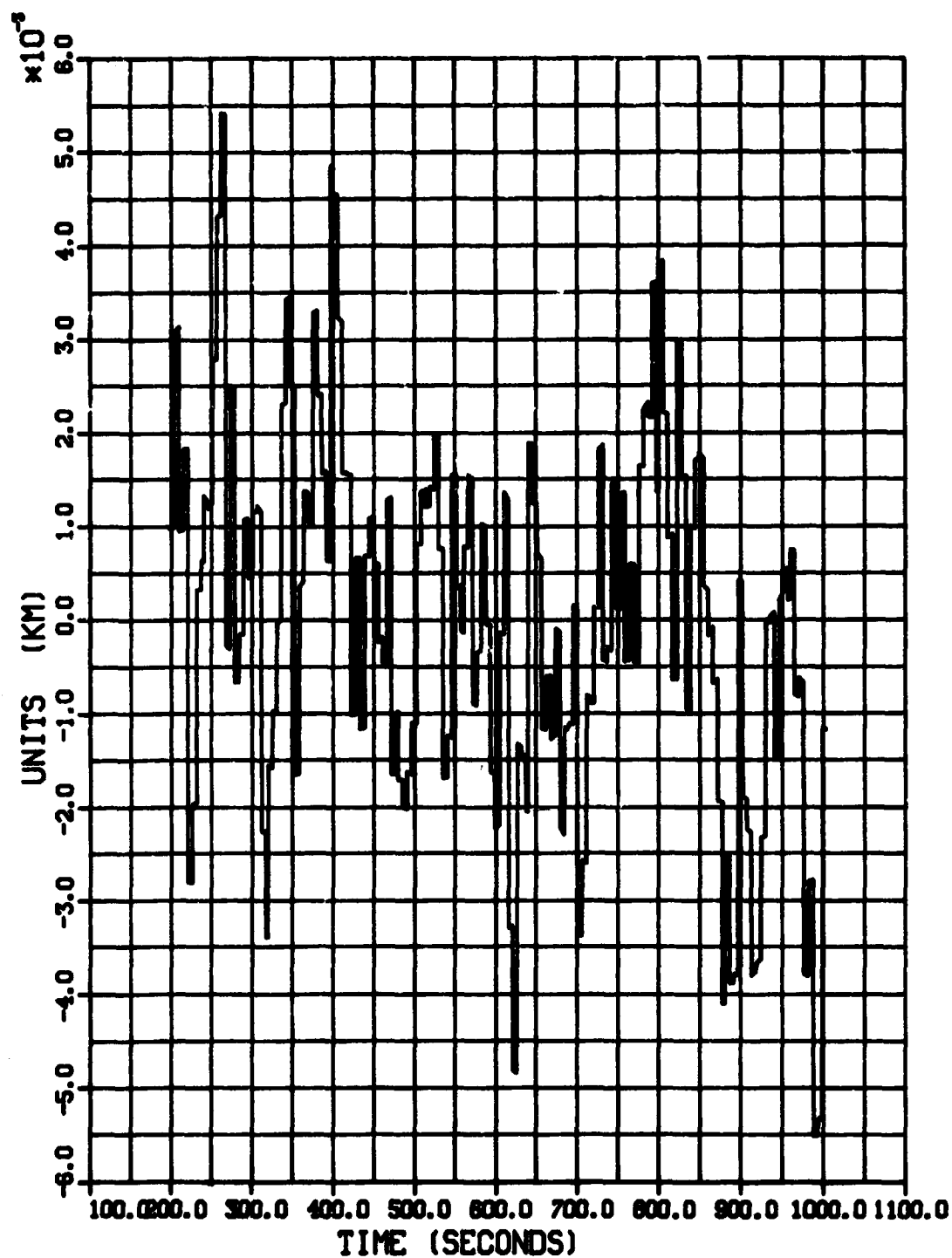
CASE10-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

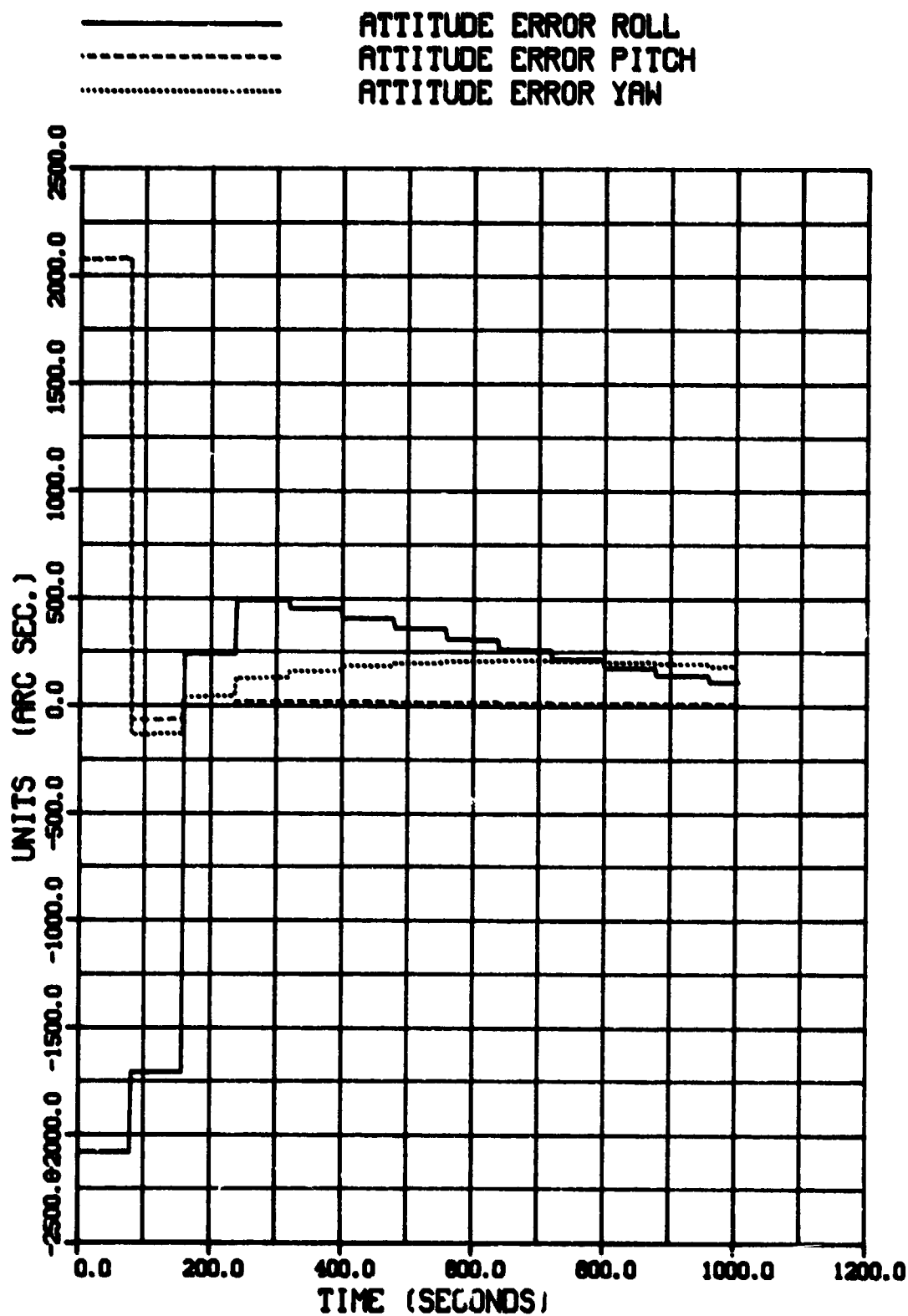


CASE10-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

NAVIGATION X POSITION ERROR

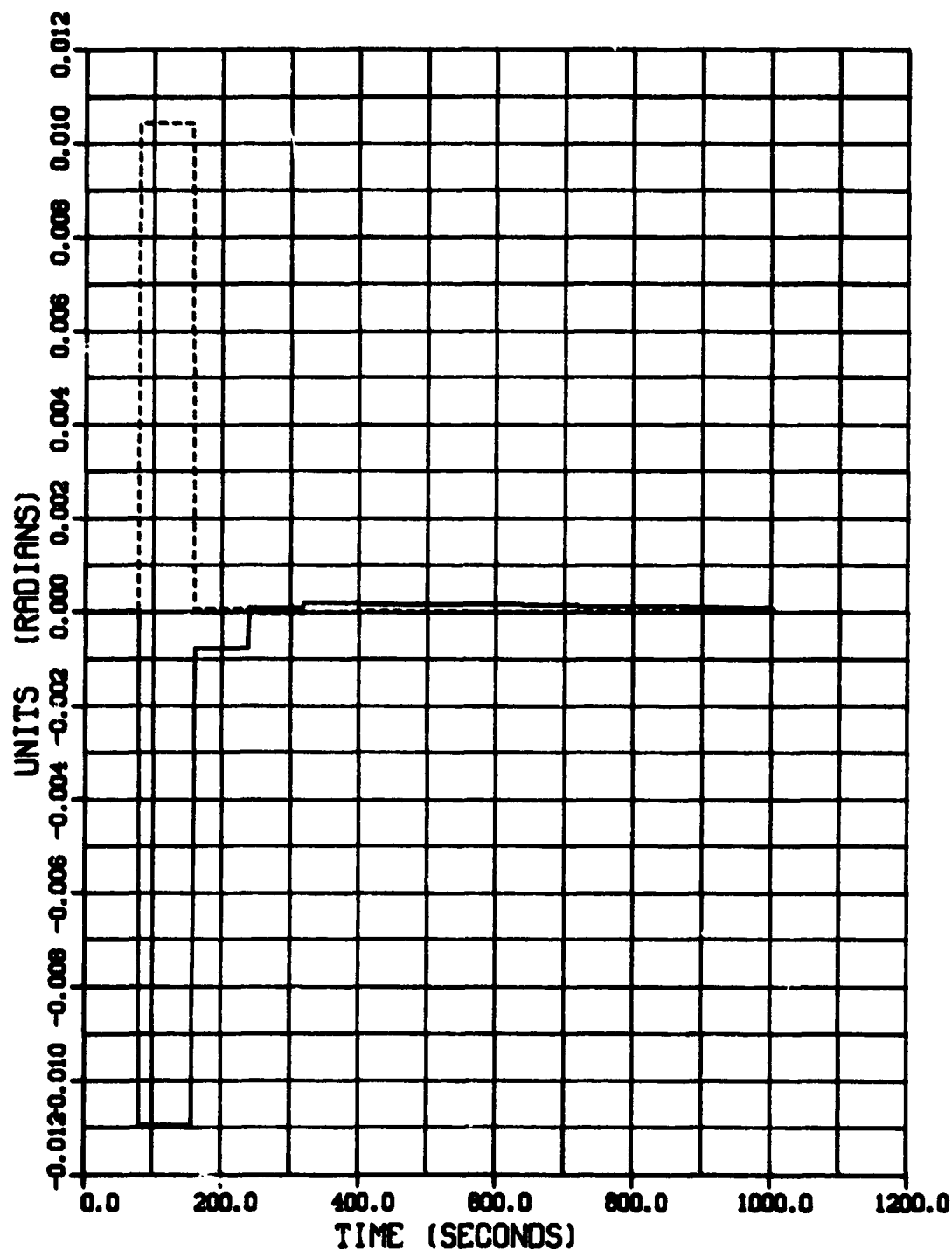


CASE10-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***



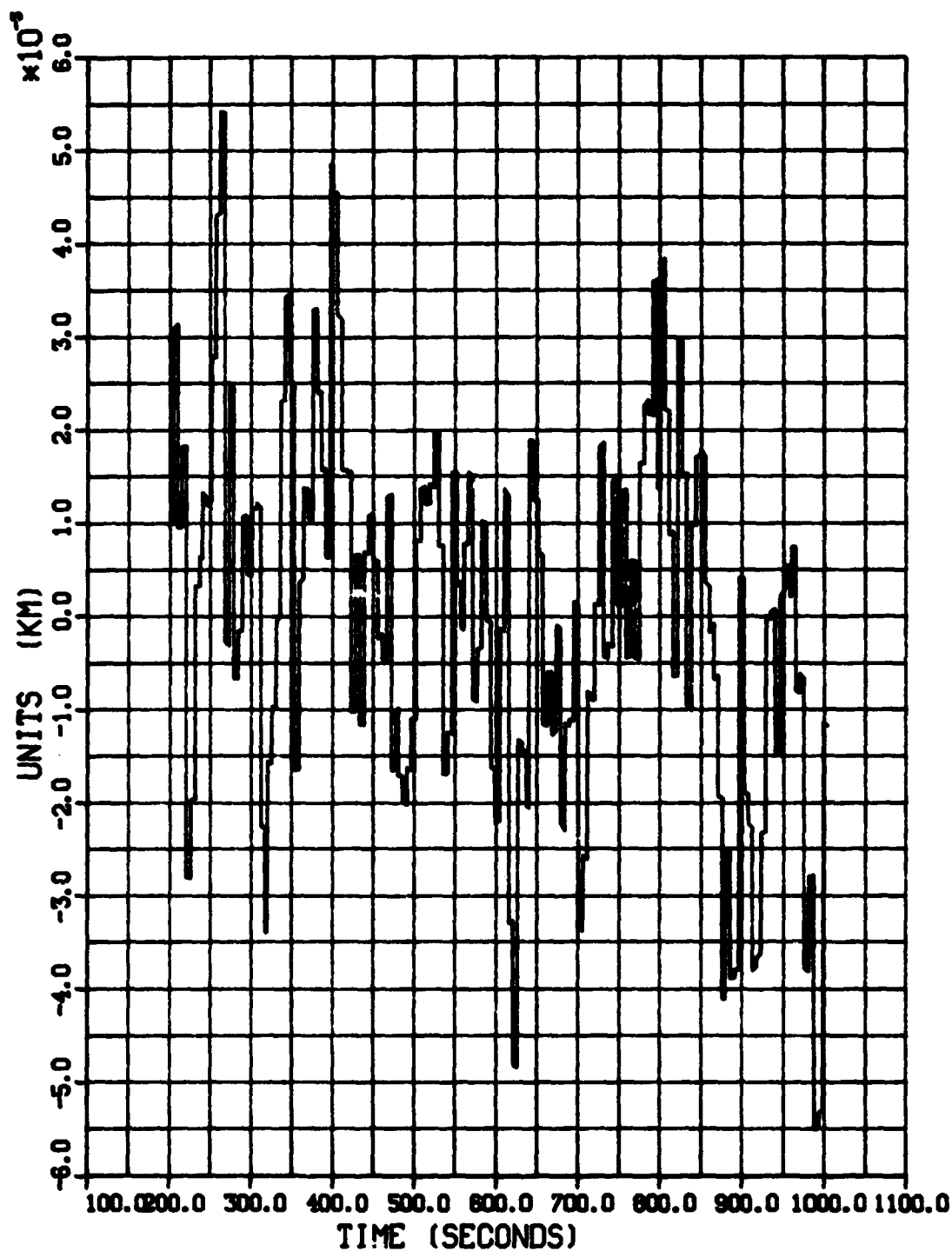
CASE10-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

—— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

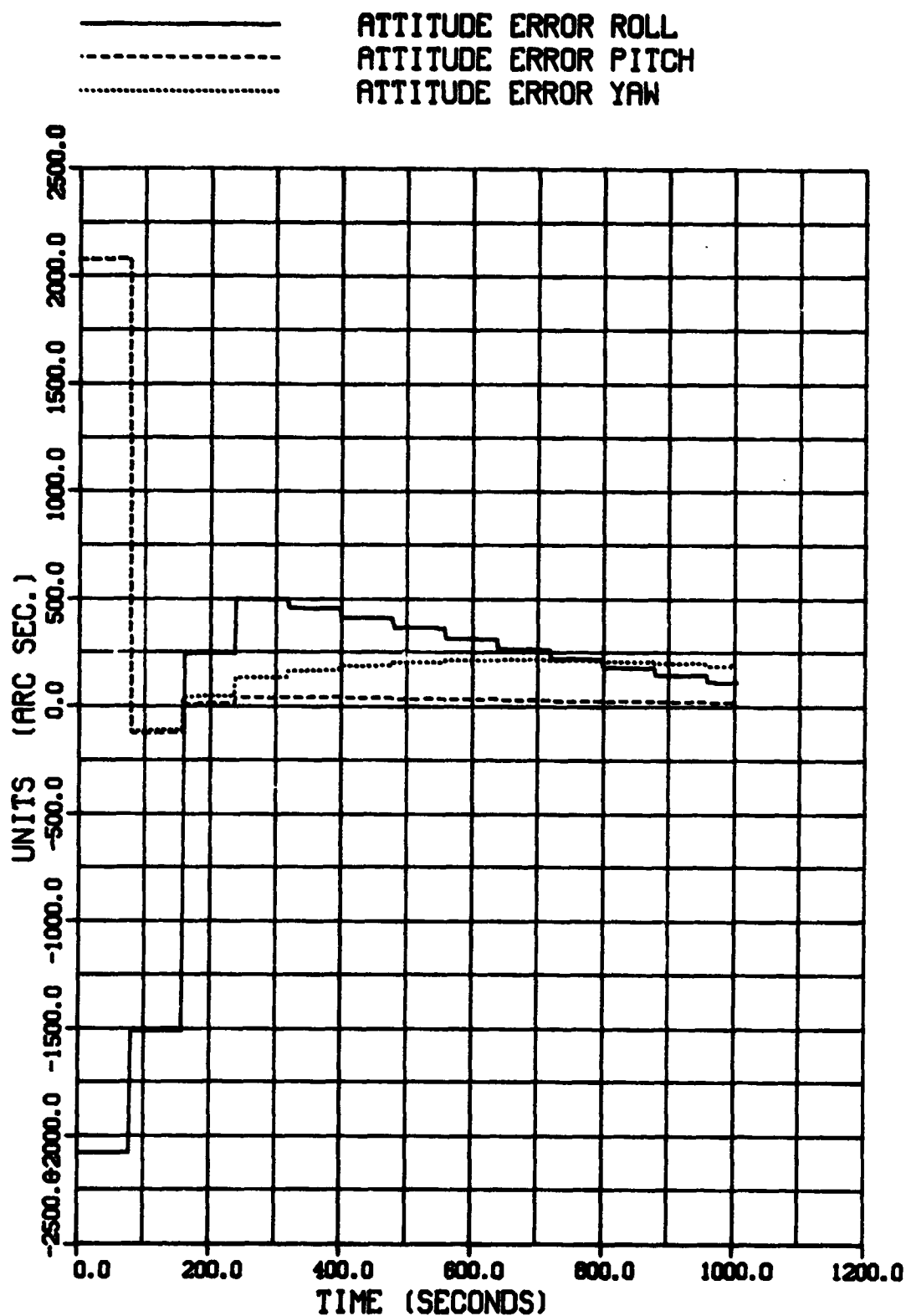


CASE12-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

NAVIGATION X POSITION ERROR



CASE12-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***



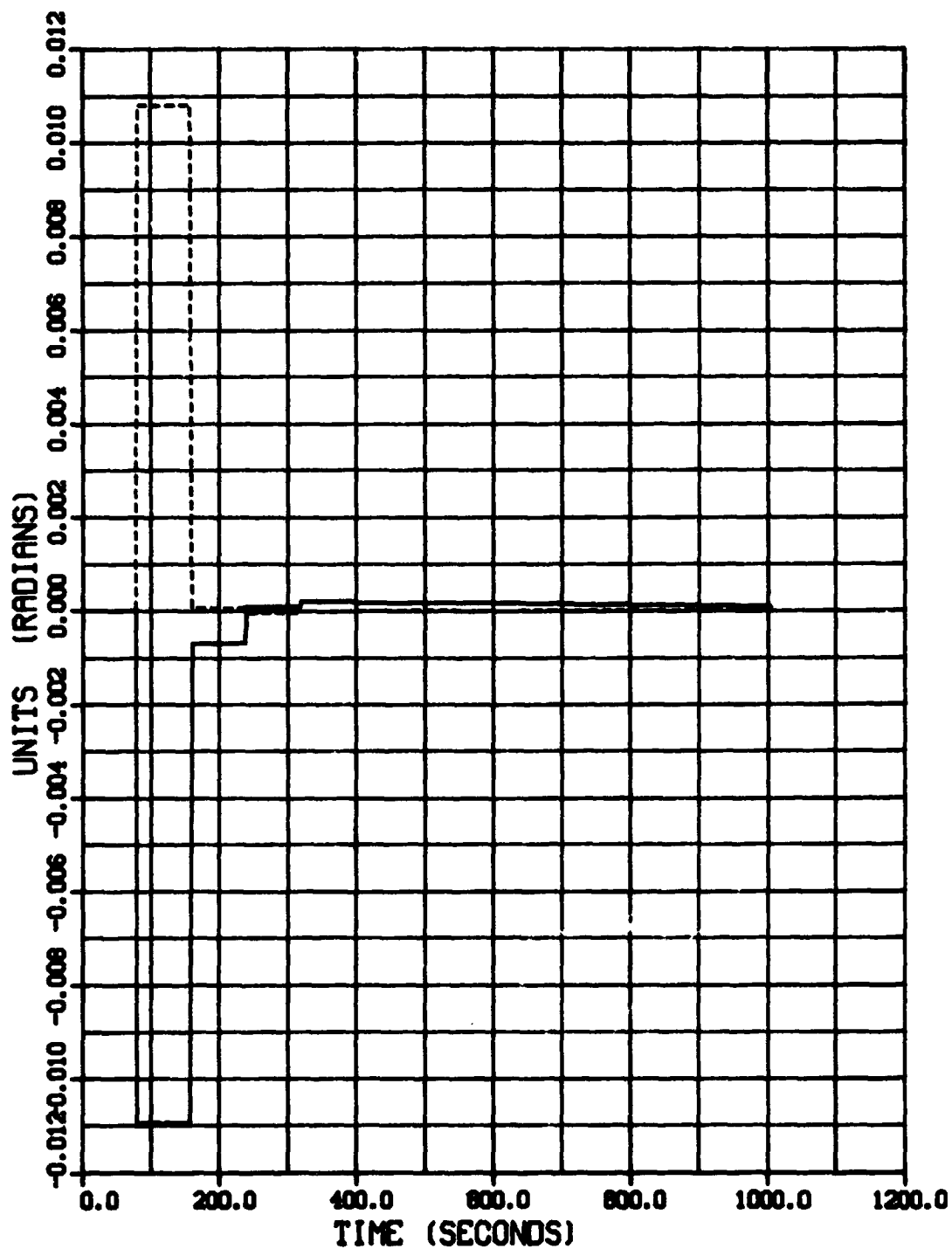
CASE12-DATA GCP-80SEC GPS-5SEC LA-FIXED

*** INERTIAL COORDINATE FRAME ***

—————
- - - - -

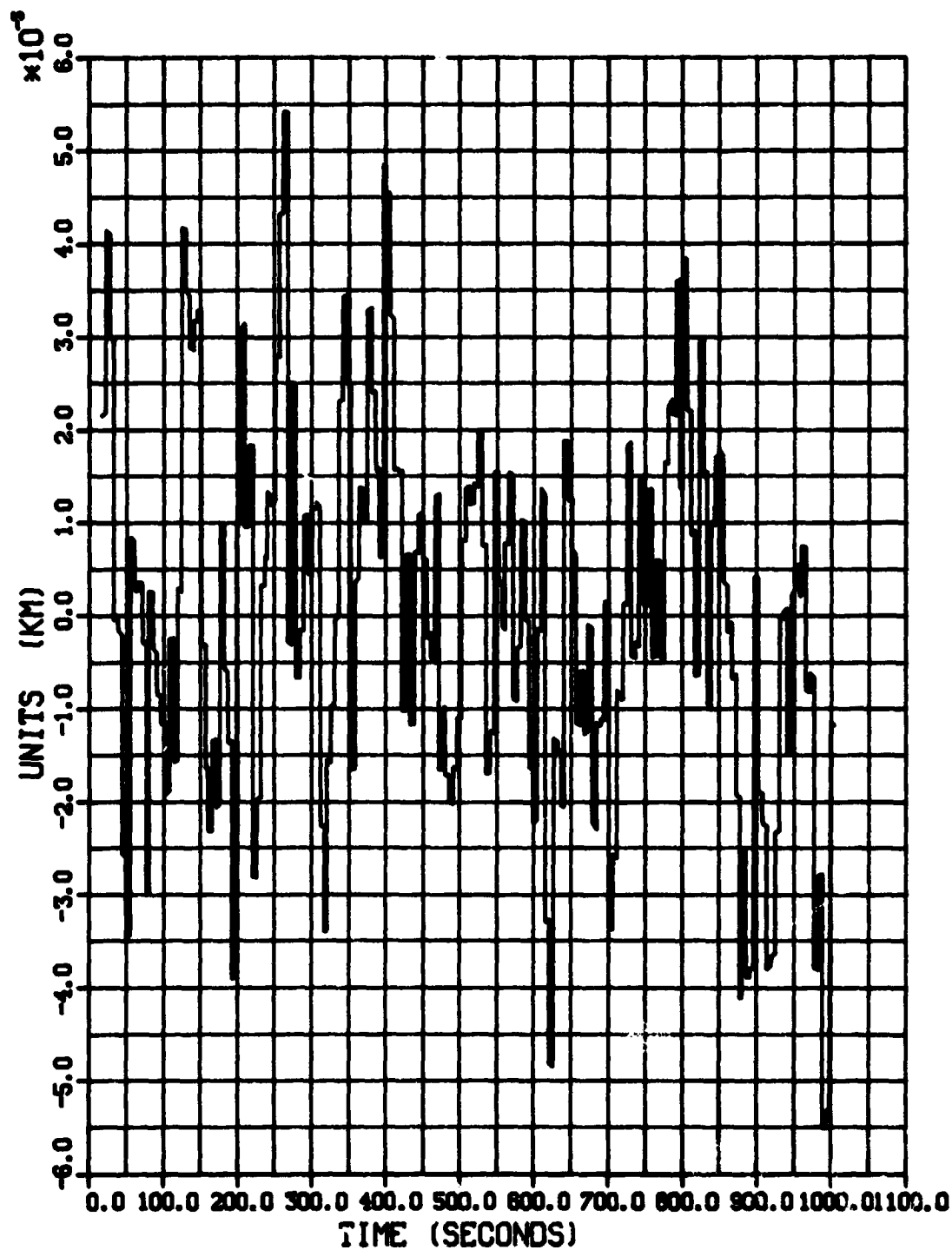
L.M. TRACKER RESIDUAL HORIZONTAL

L.M. TRACKER RESIDUAL VERTICAL

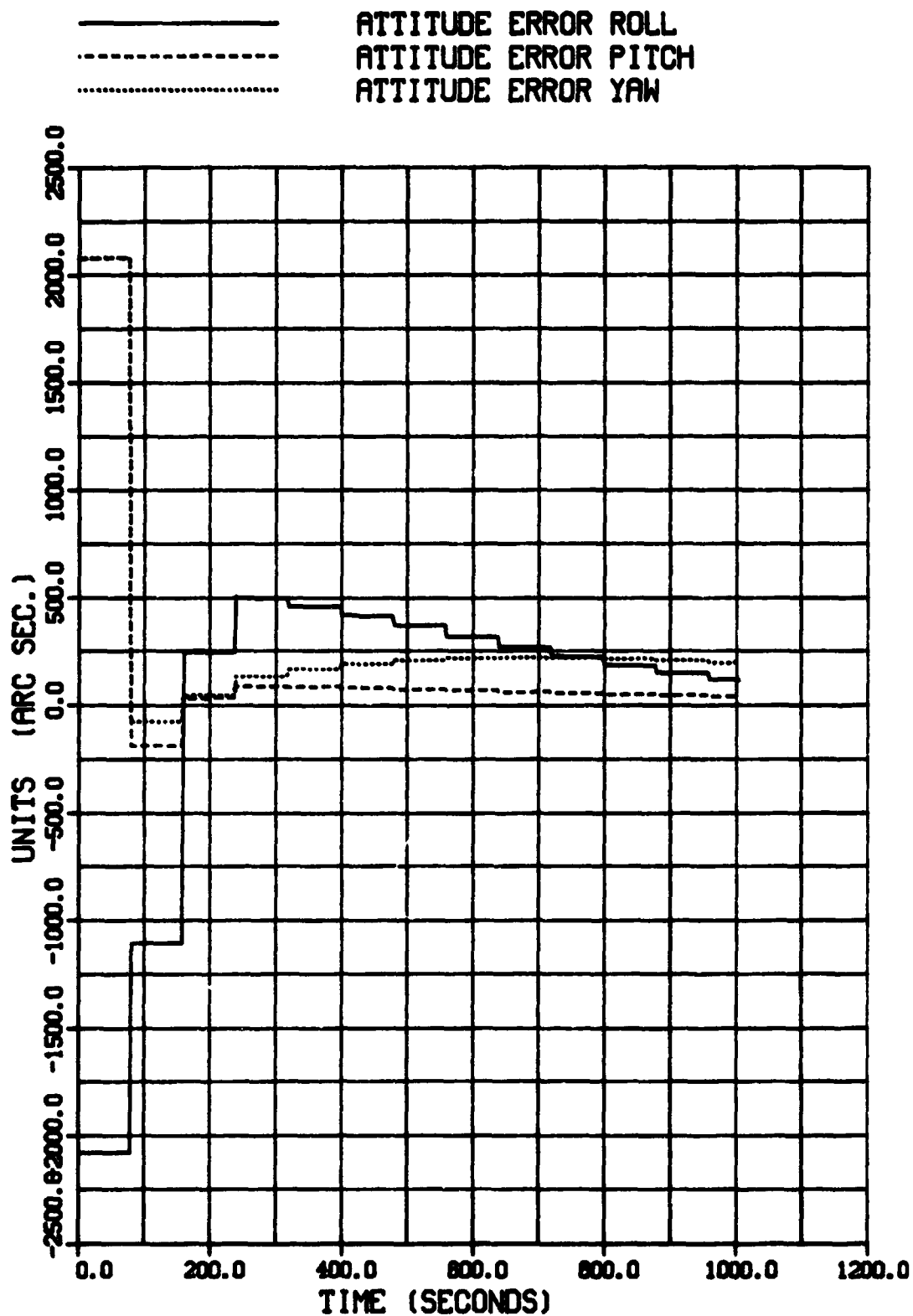


CASE14-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

NAVIGATION X POSITION ERROR

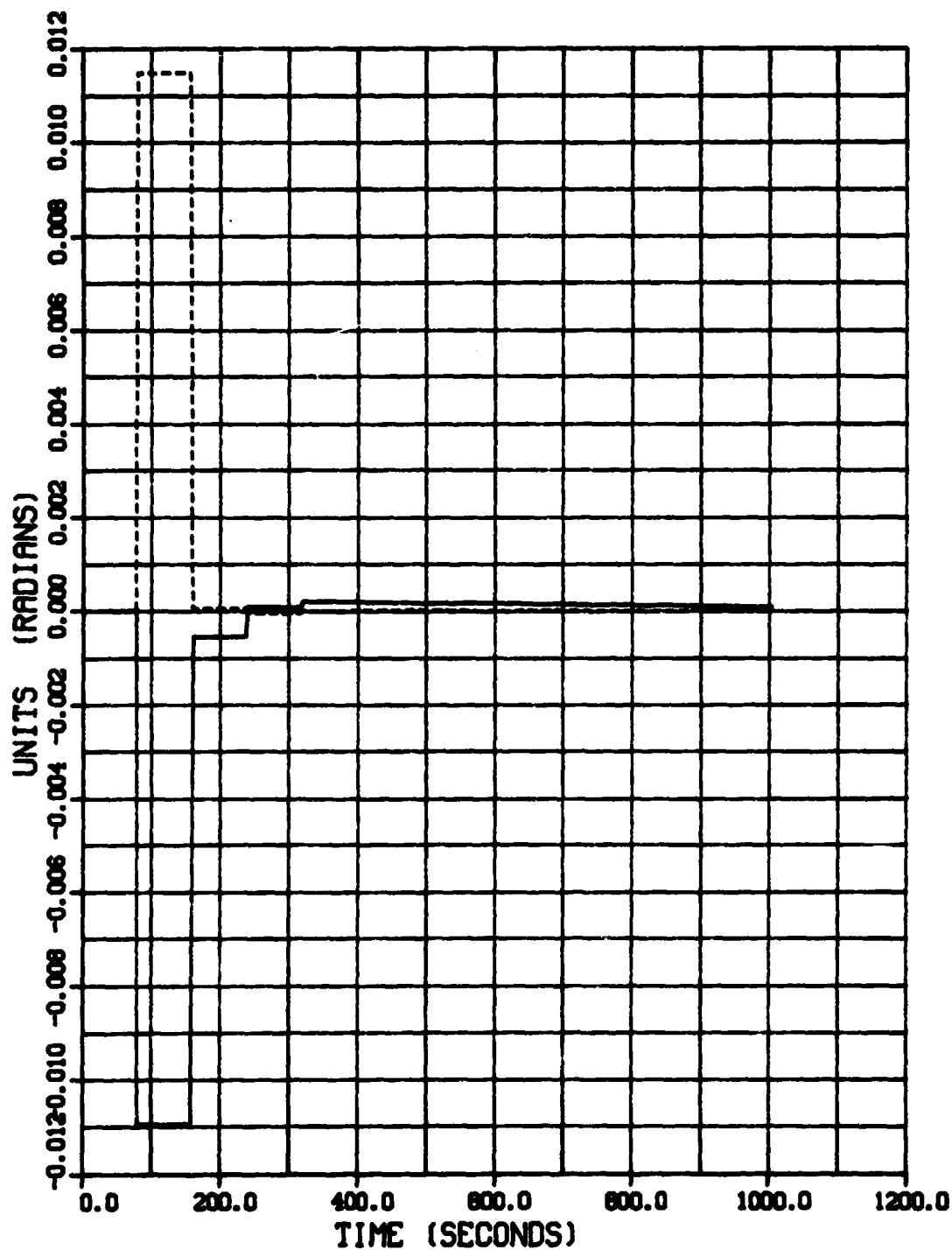


CASE14-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***



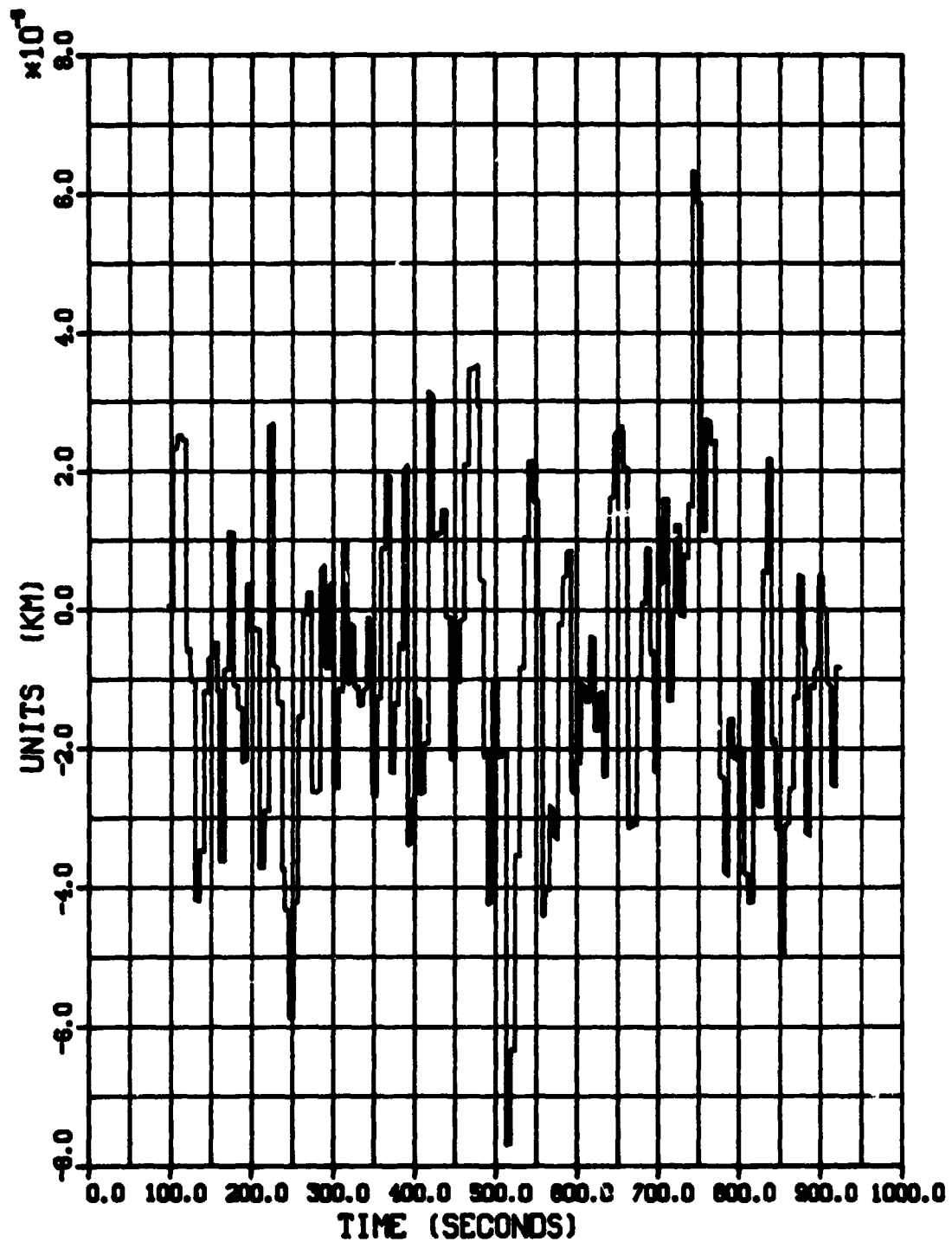
CASE14-DATA GCP-80SEC GPS-5SEC LA-FIXED
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

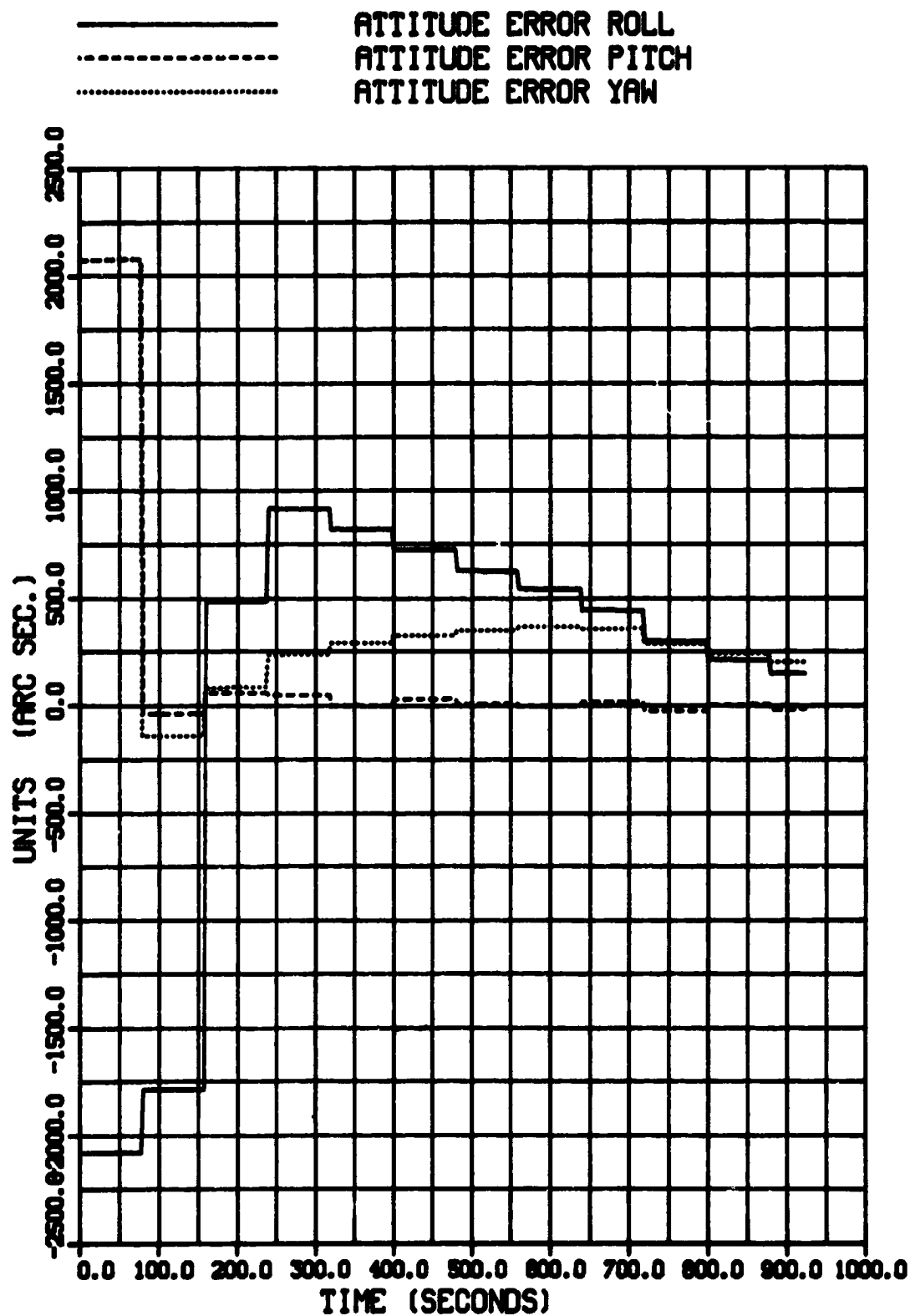


CASE17-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

NAVIGATION X POSITION ERROR

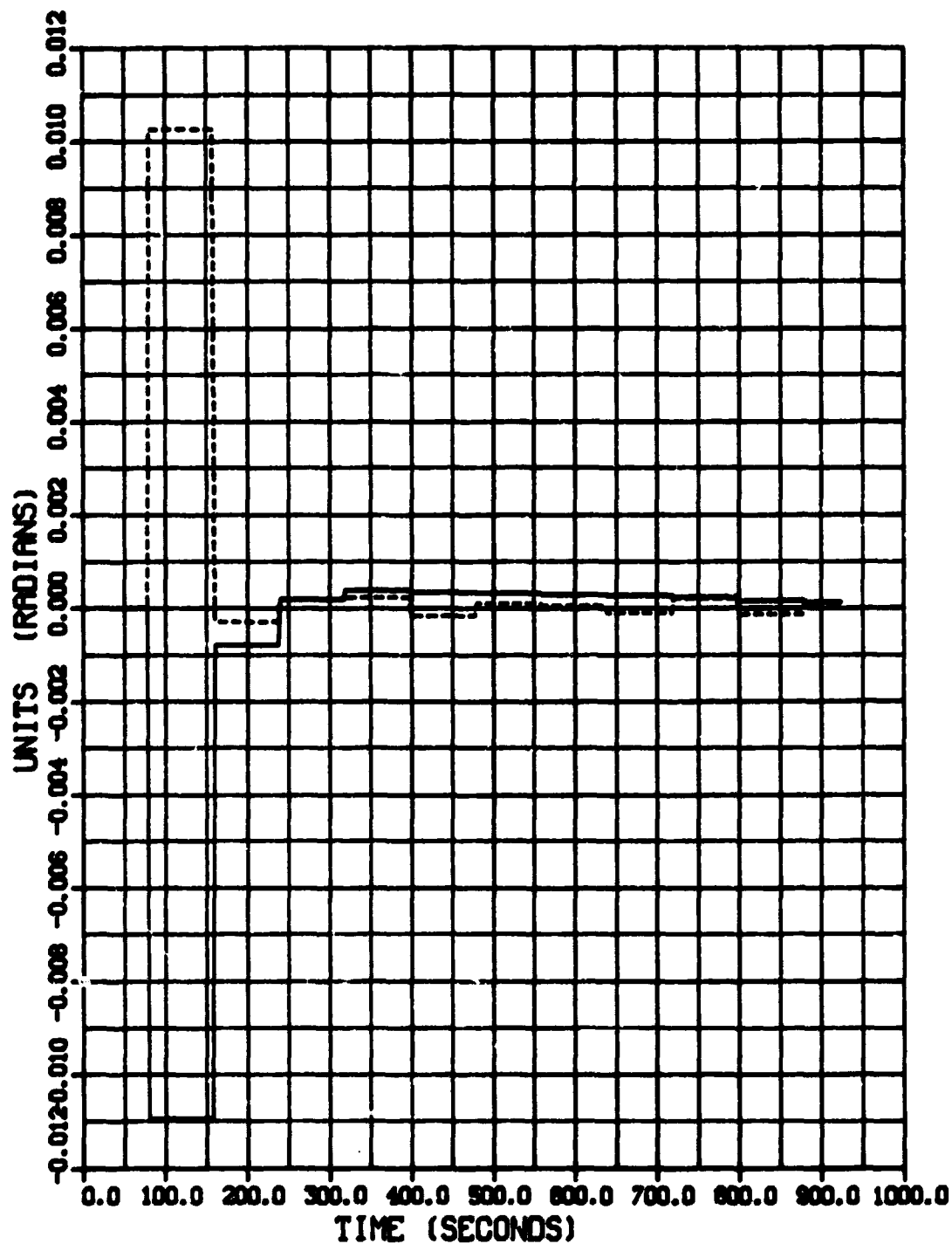


CASE17-DATA GCP-80SEC GPS-5SEC LA-RANDO
 *** INERTIAL COORDINATE FRAME ***



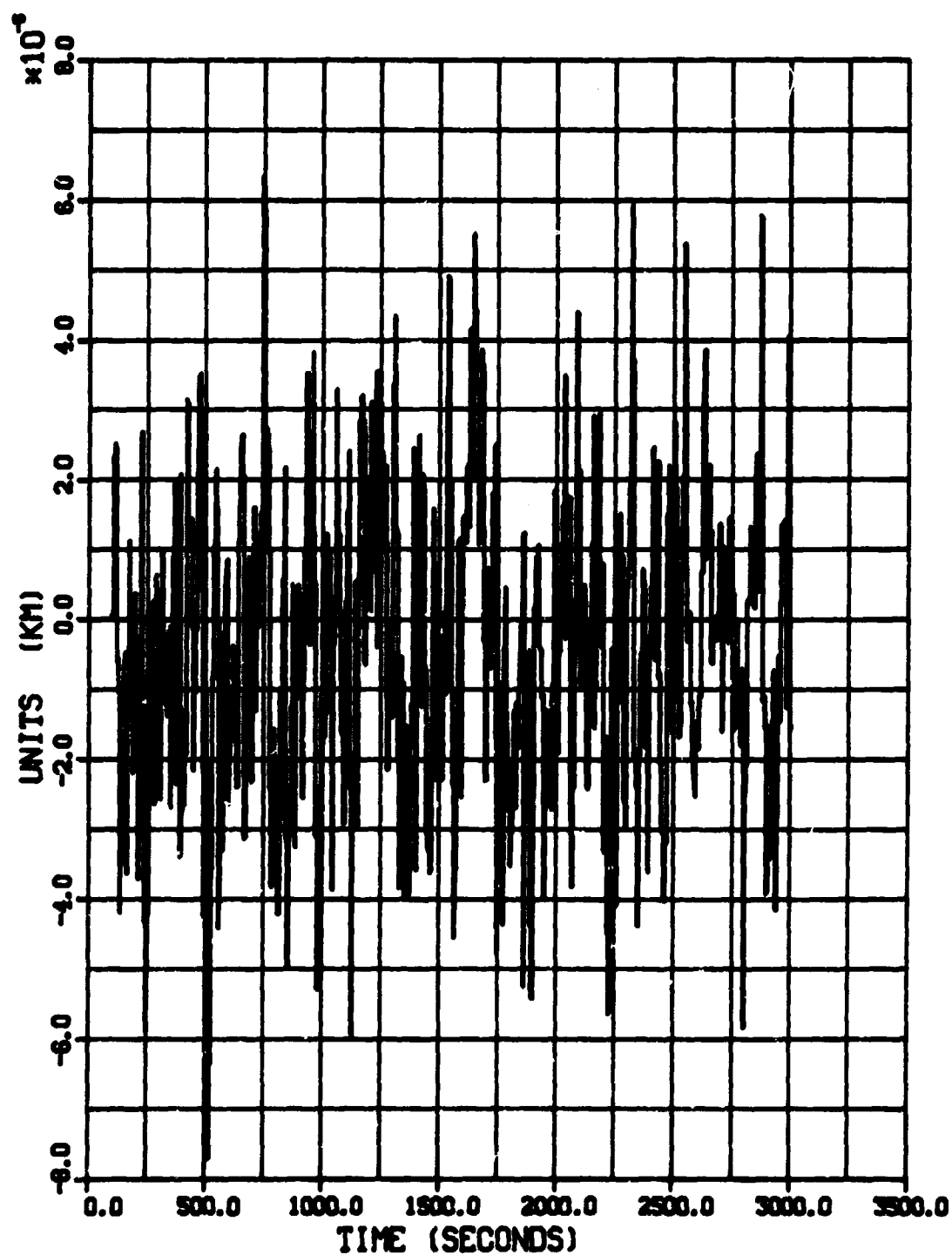
CASE17-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

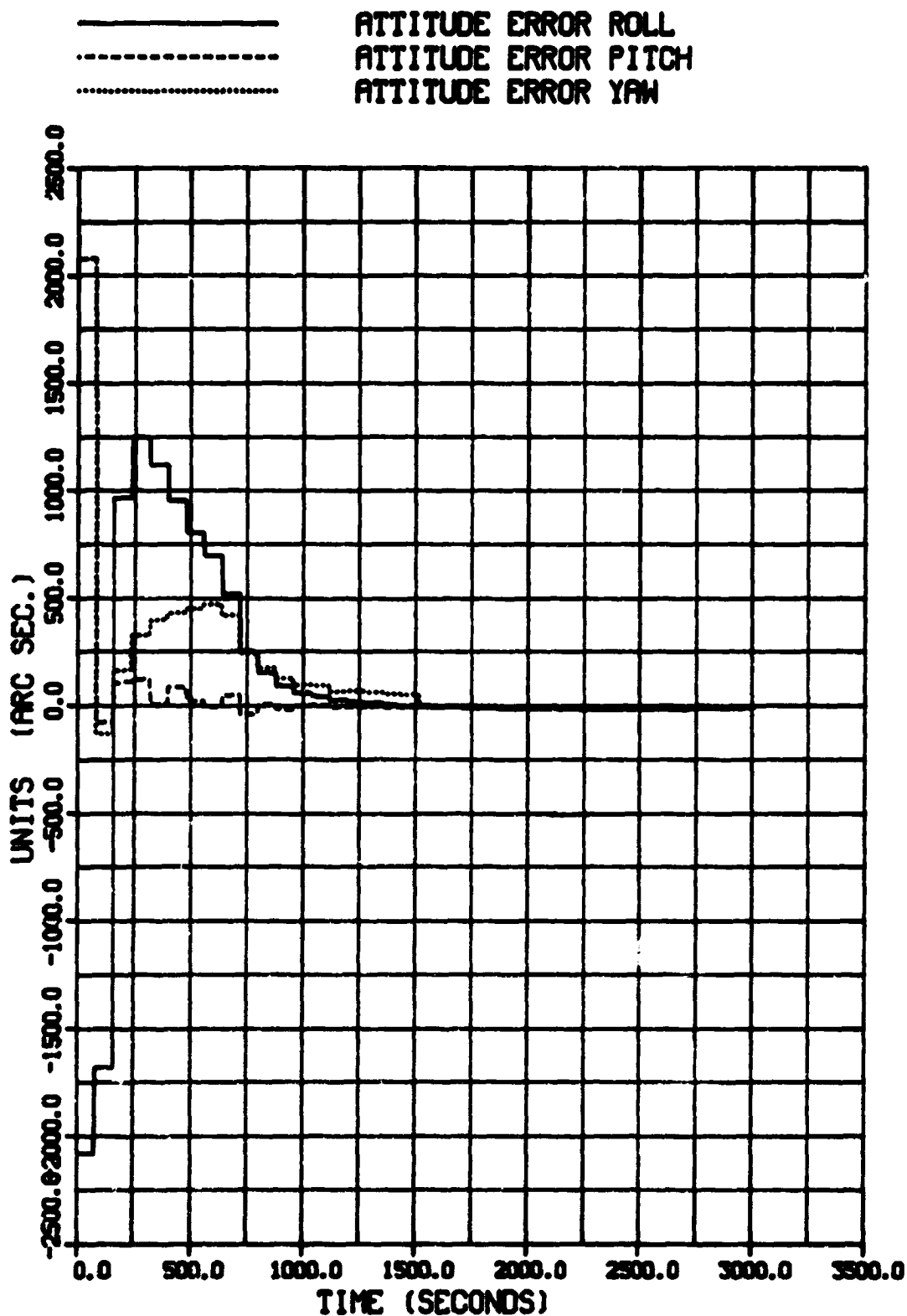


CASE18-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

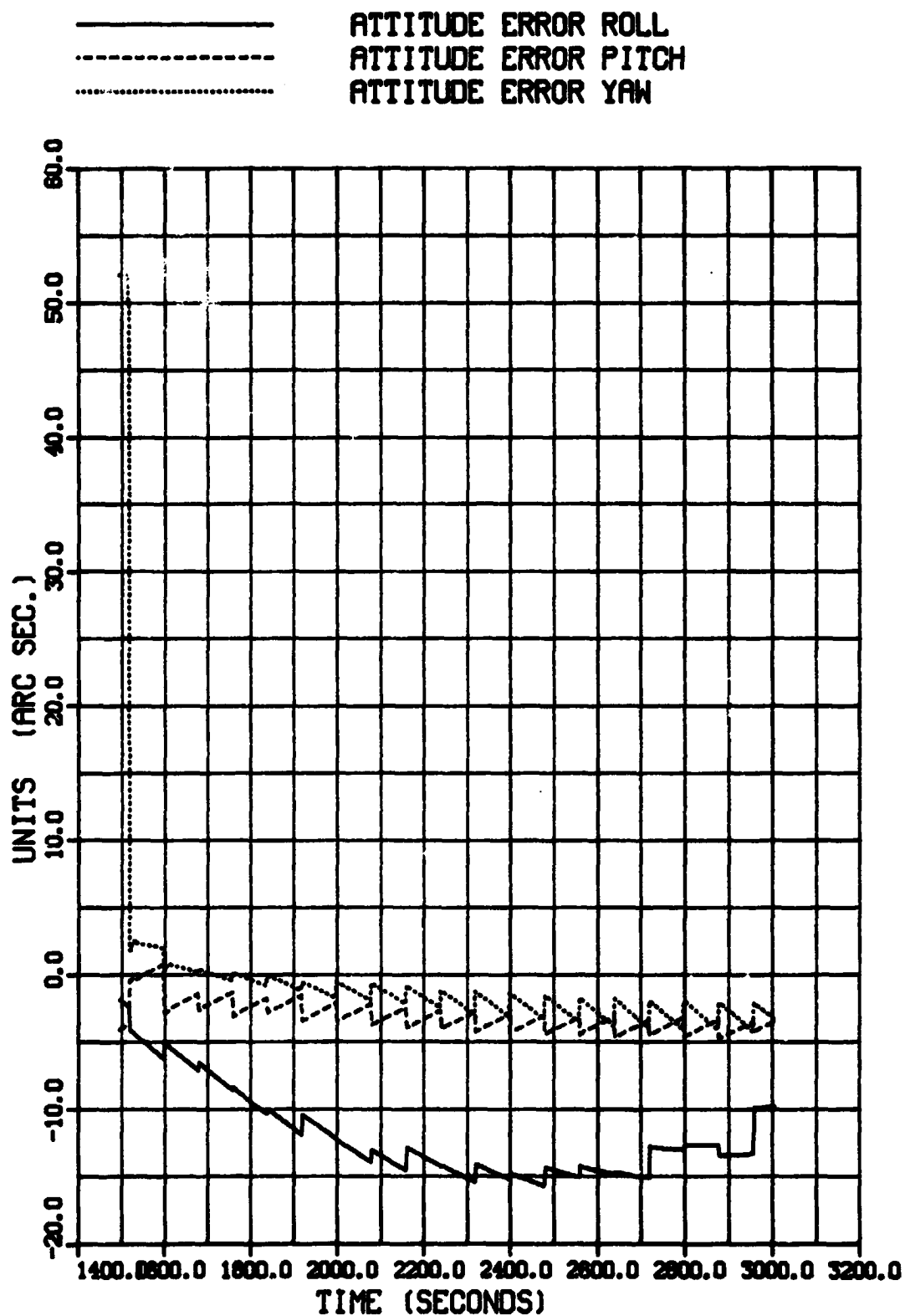
NAVIGATION X POSITION ERROR



CASE18-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***



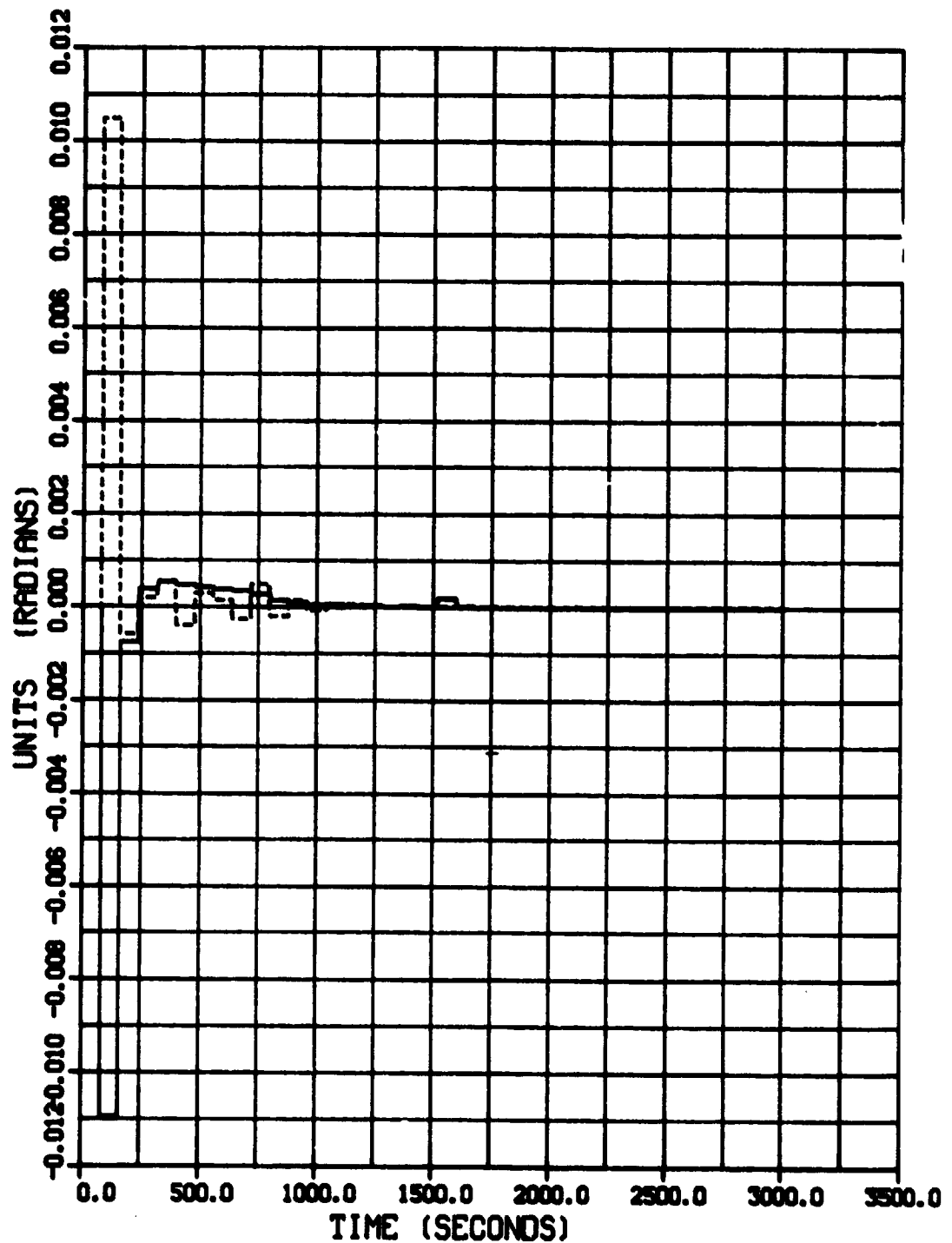
CASE18-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***



CASE18-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

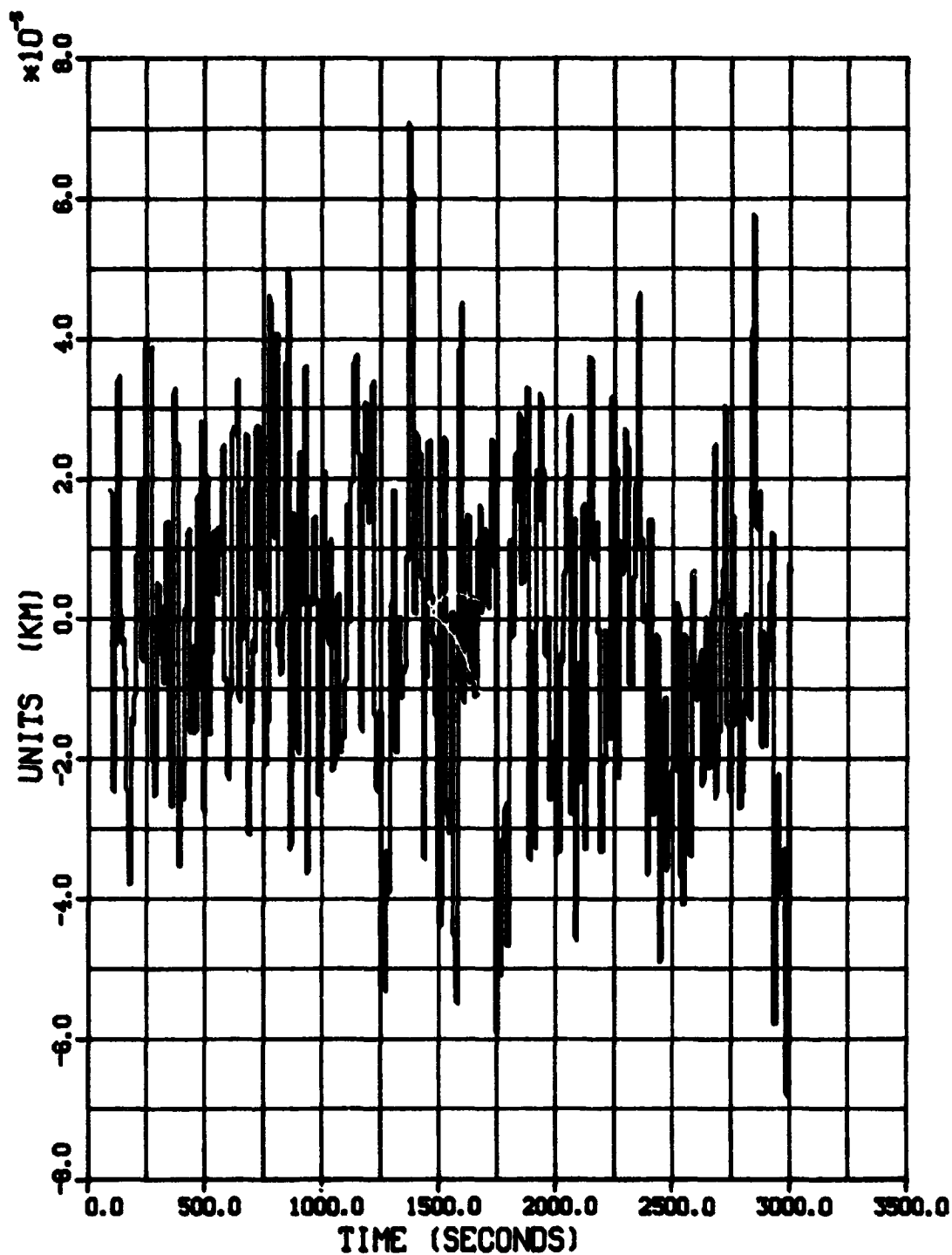
—————
- - - - -

L.M. TRACKER RESIDUAL HORIZONTAL
L.M. TRACKER RESIDUAL VERTICAL

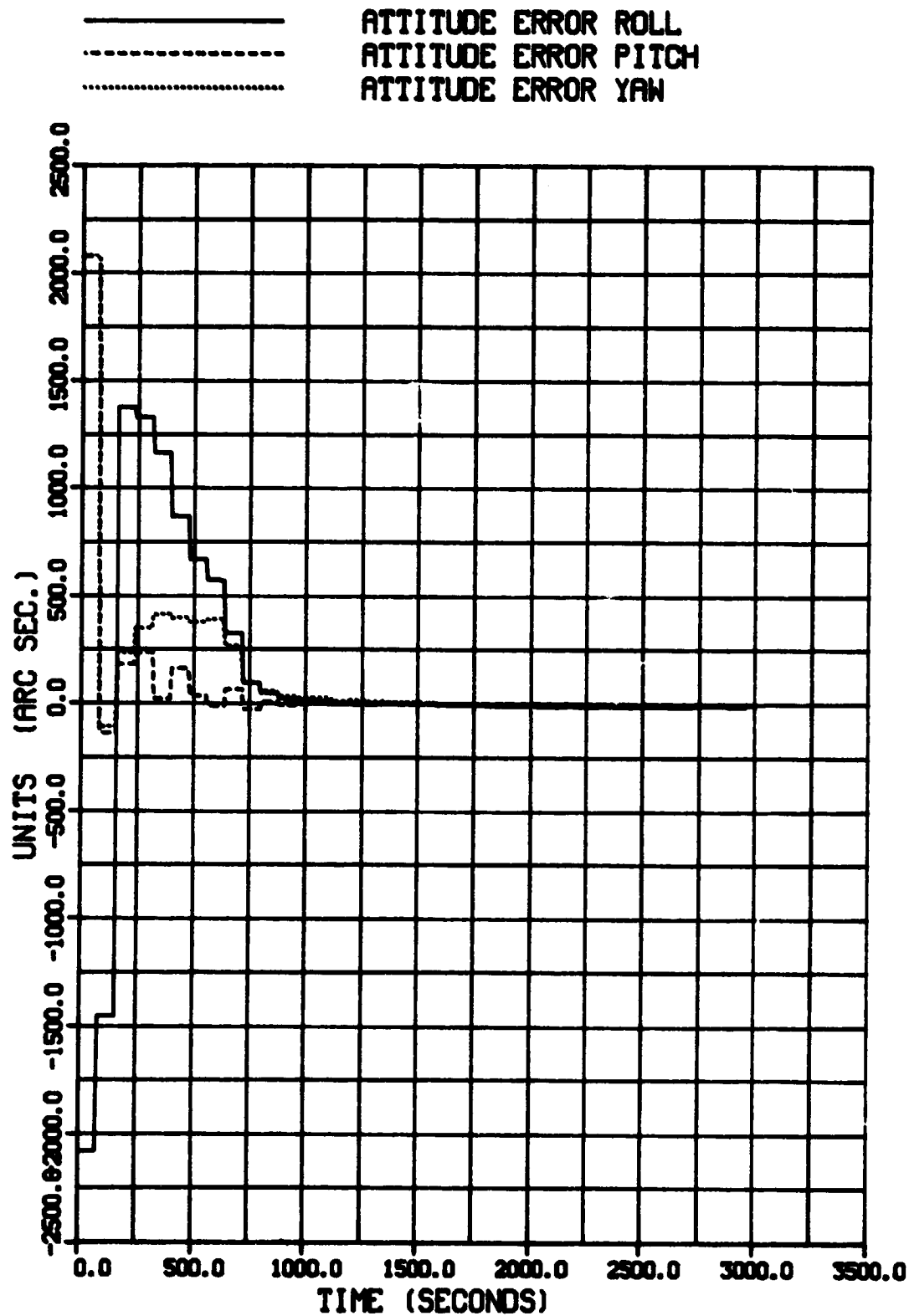


CASE21-DATA GCP-80SEC GPS-10SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

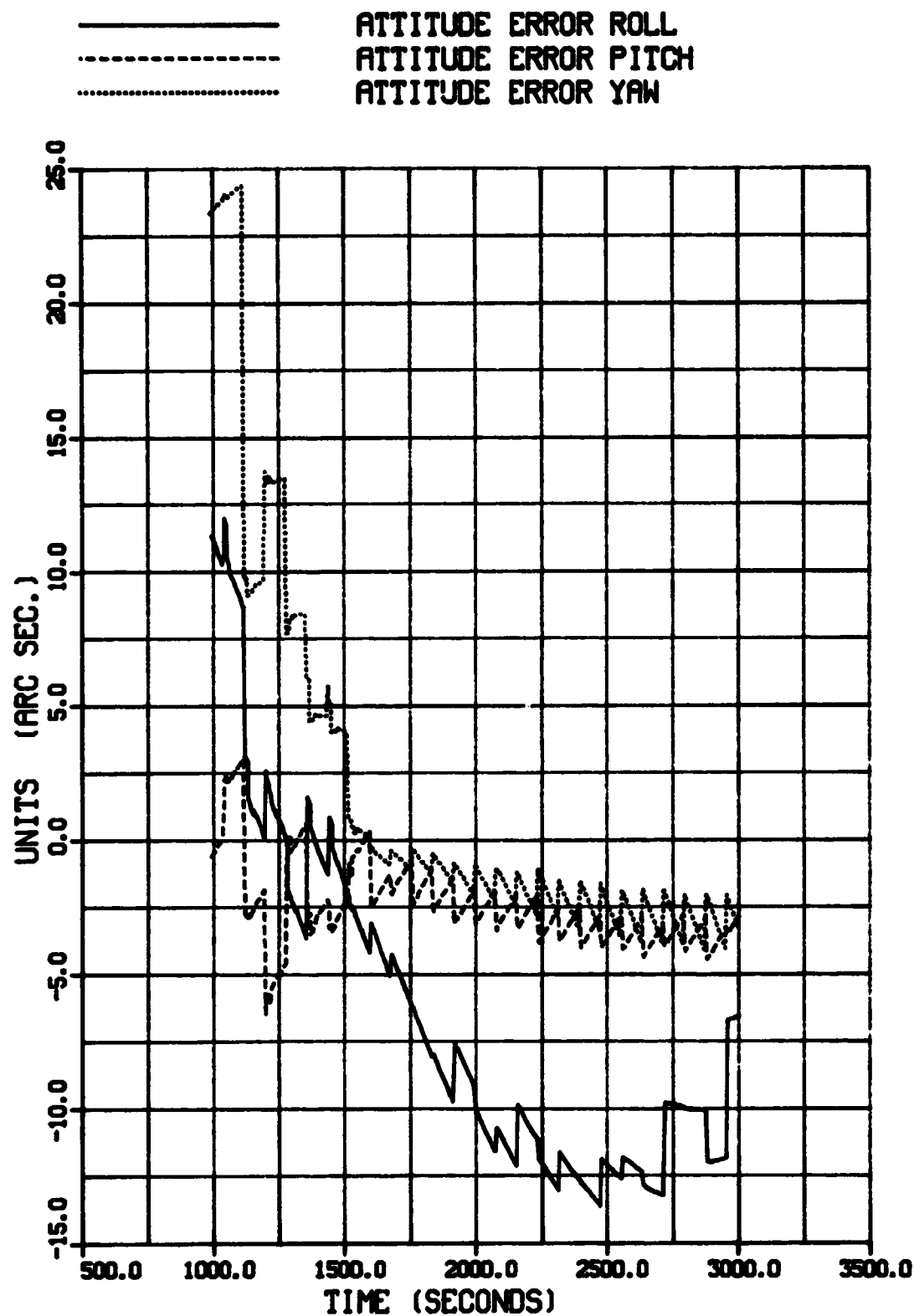
NAVIGATION X POSITION ERROR



CASE21-DATA GCP-80SEC GPS-10SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

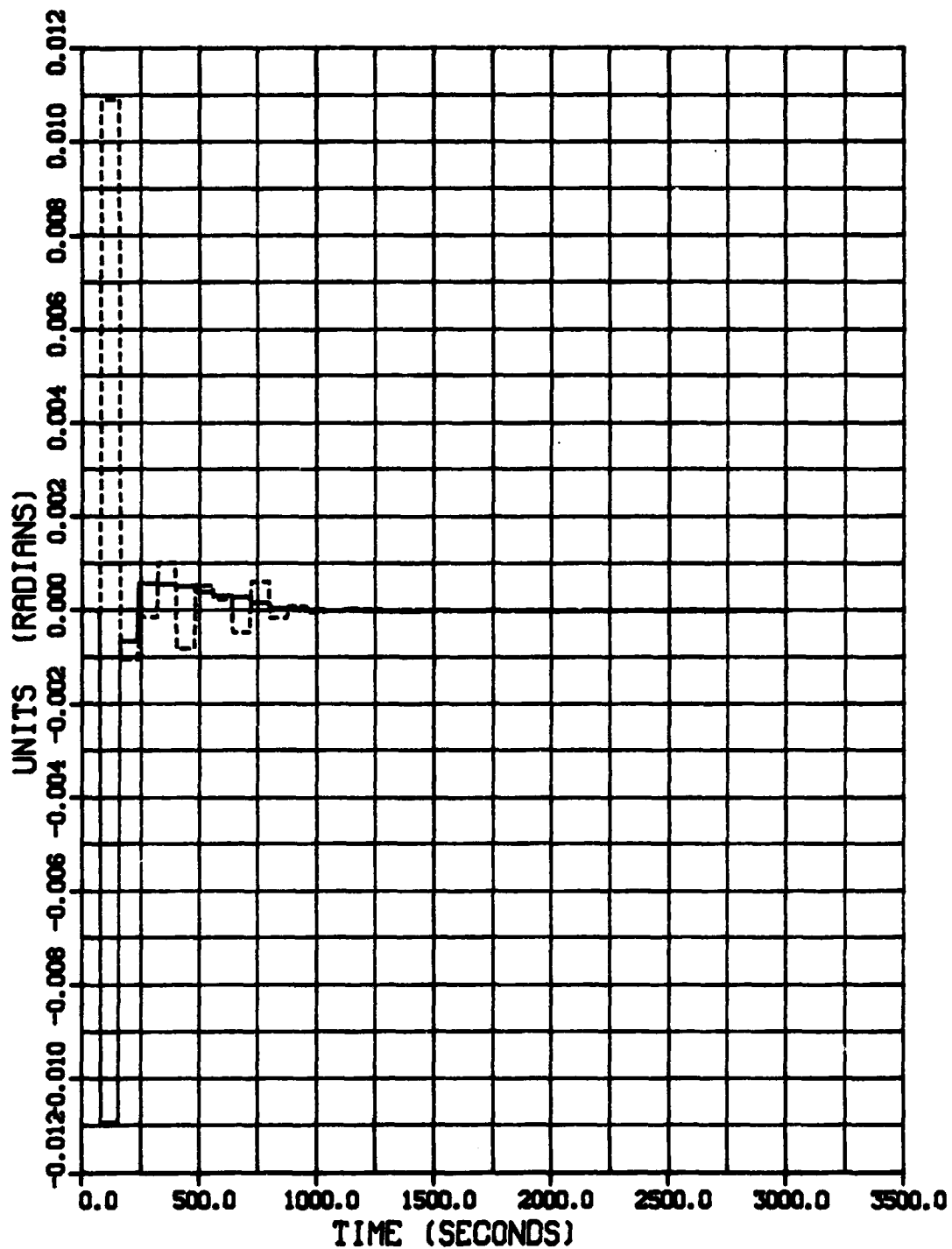


CASE21-DATA GCP-80SEC GPS-10SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***



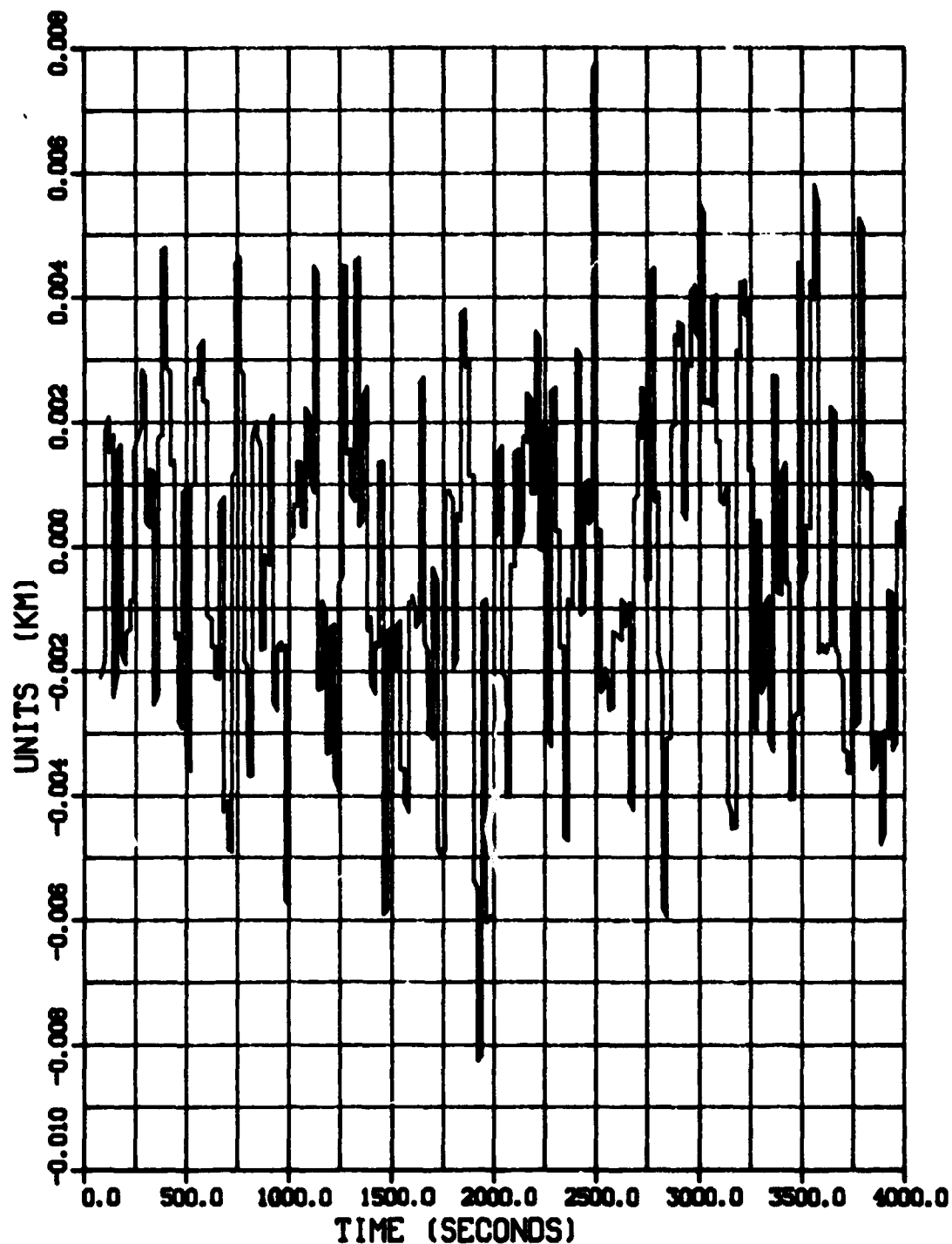
CASE21-DATA GCP-80SEC GPS-10SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

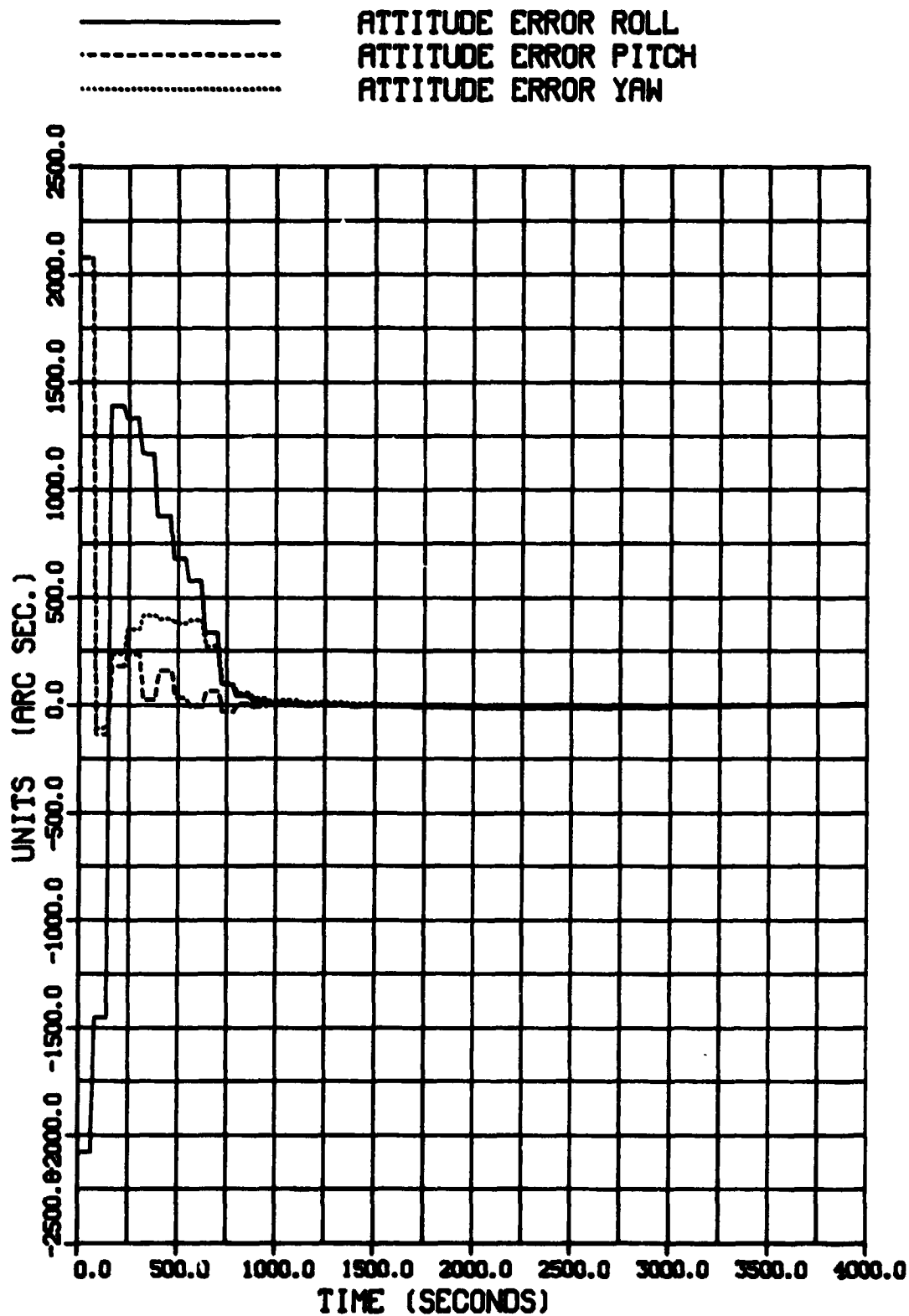


CASE22-DATA GCP-80SEC GPS-20SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

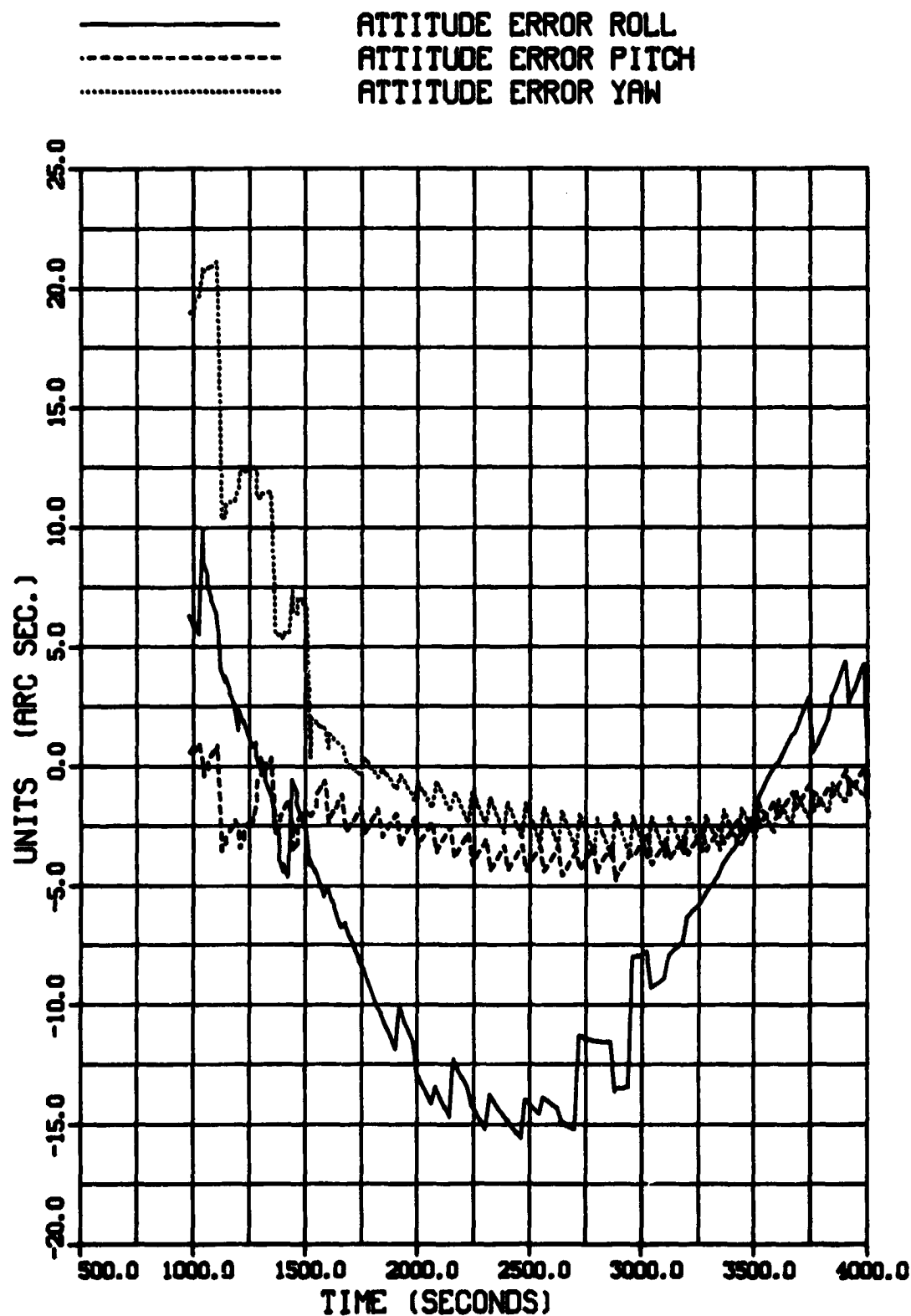
NAVIGATION X POSITION ERROR



CASE22-DATA GCP-80SEC GPS-20SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

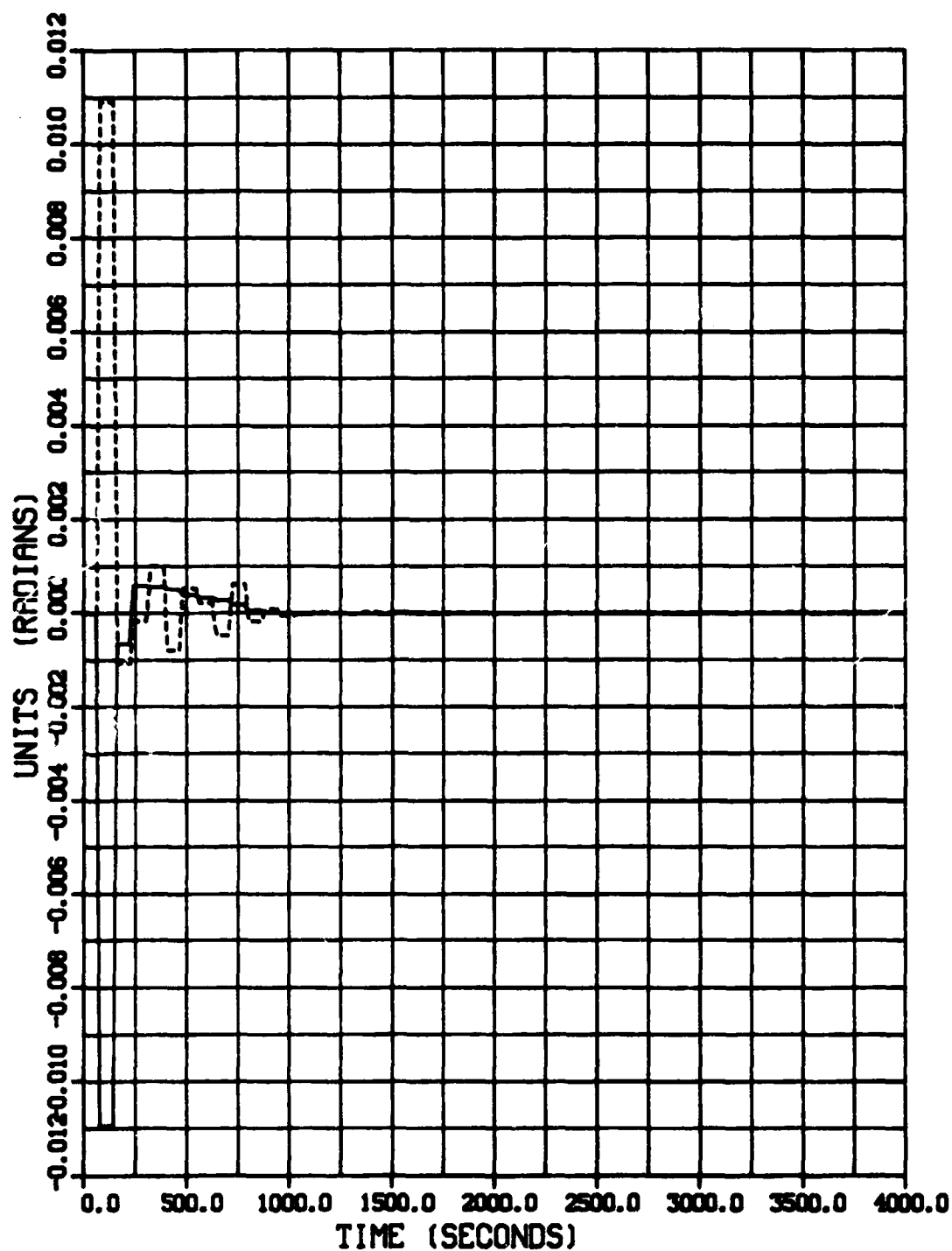


CASE22-DATA GCP-80SEC GPS-20SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***



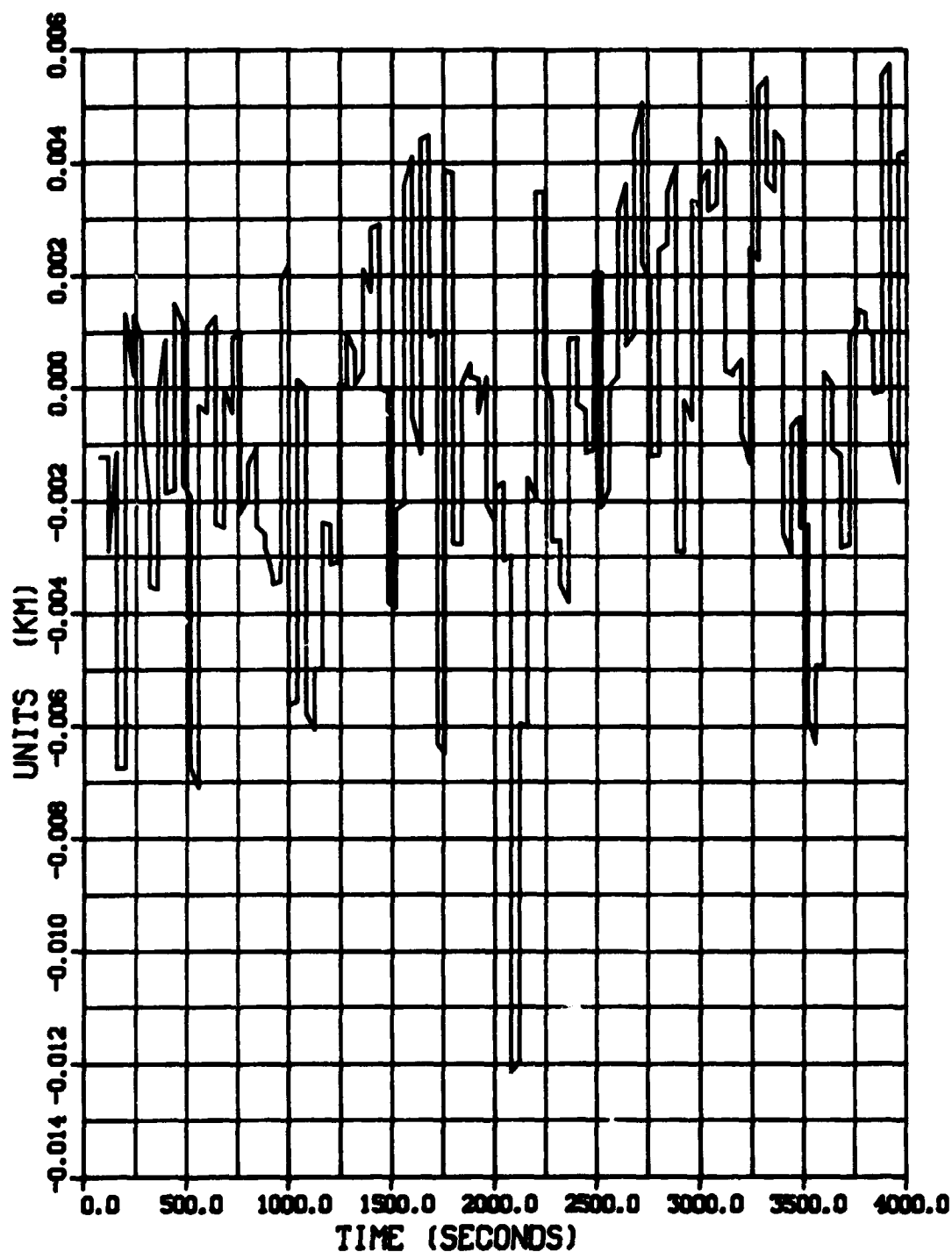
CASE22-DATA GCP-80SEC GPS-20SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

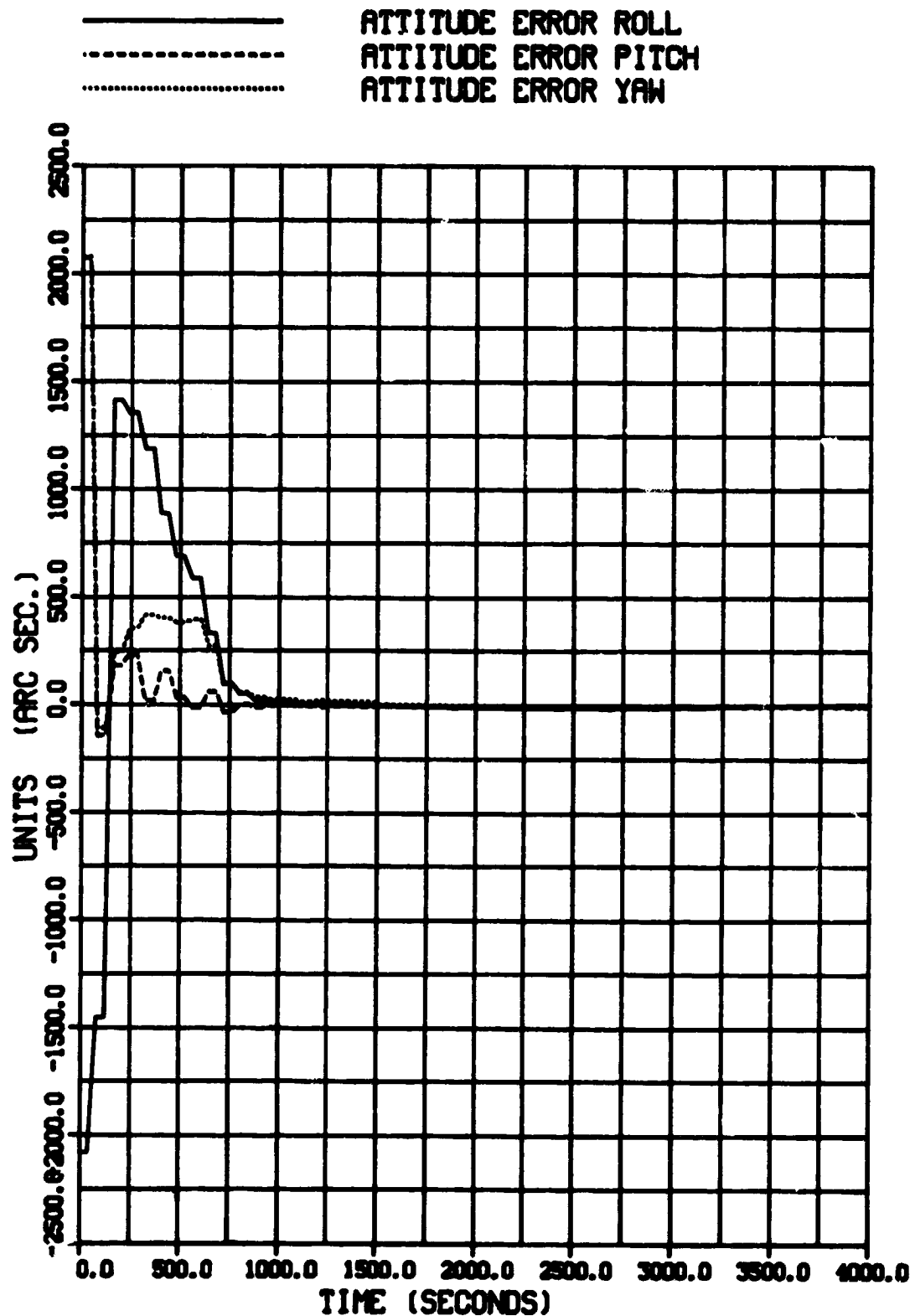


CASE23-DATA GCP-80SEC GPS-40SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

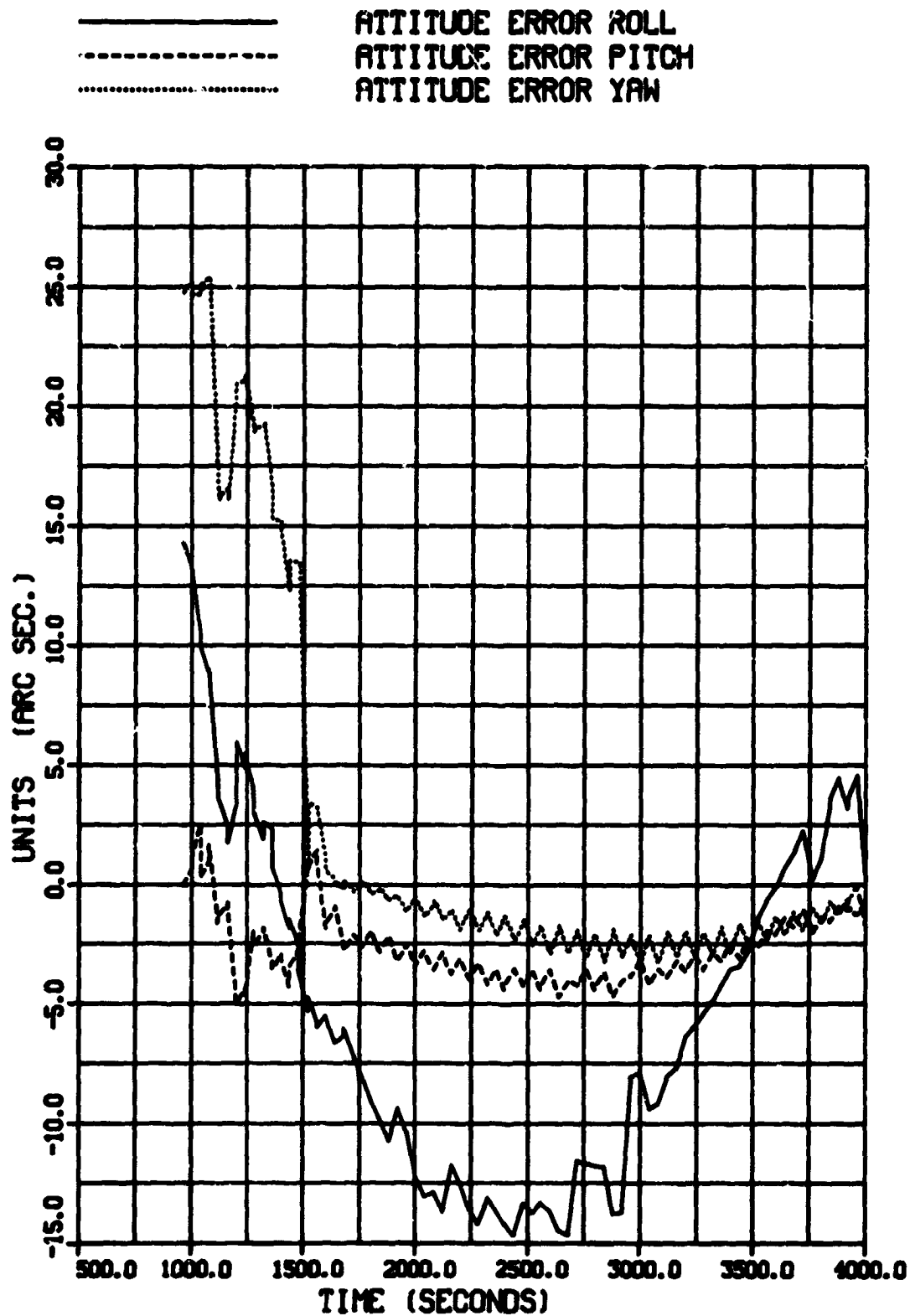
NAVIGATION X POSITION ERROR



CASE23-DATA GCP-80SEC GPS-40SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

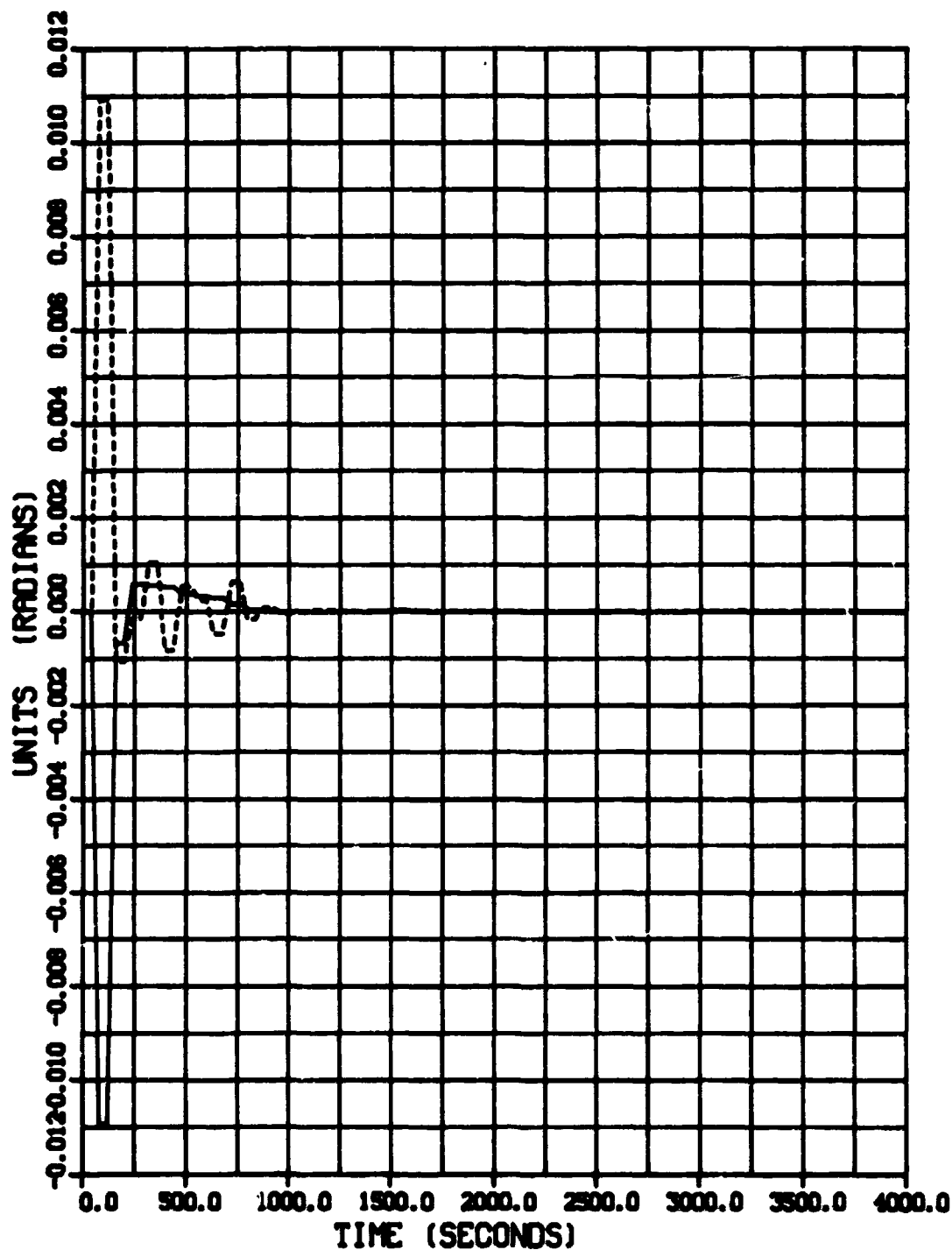


CASE23-DATA GCP-80SEC GPS-40SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***



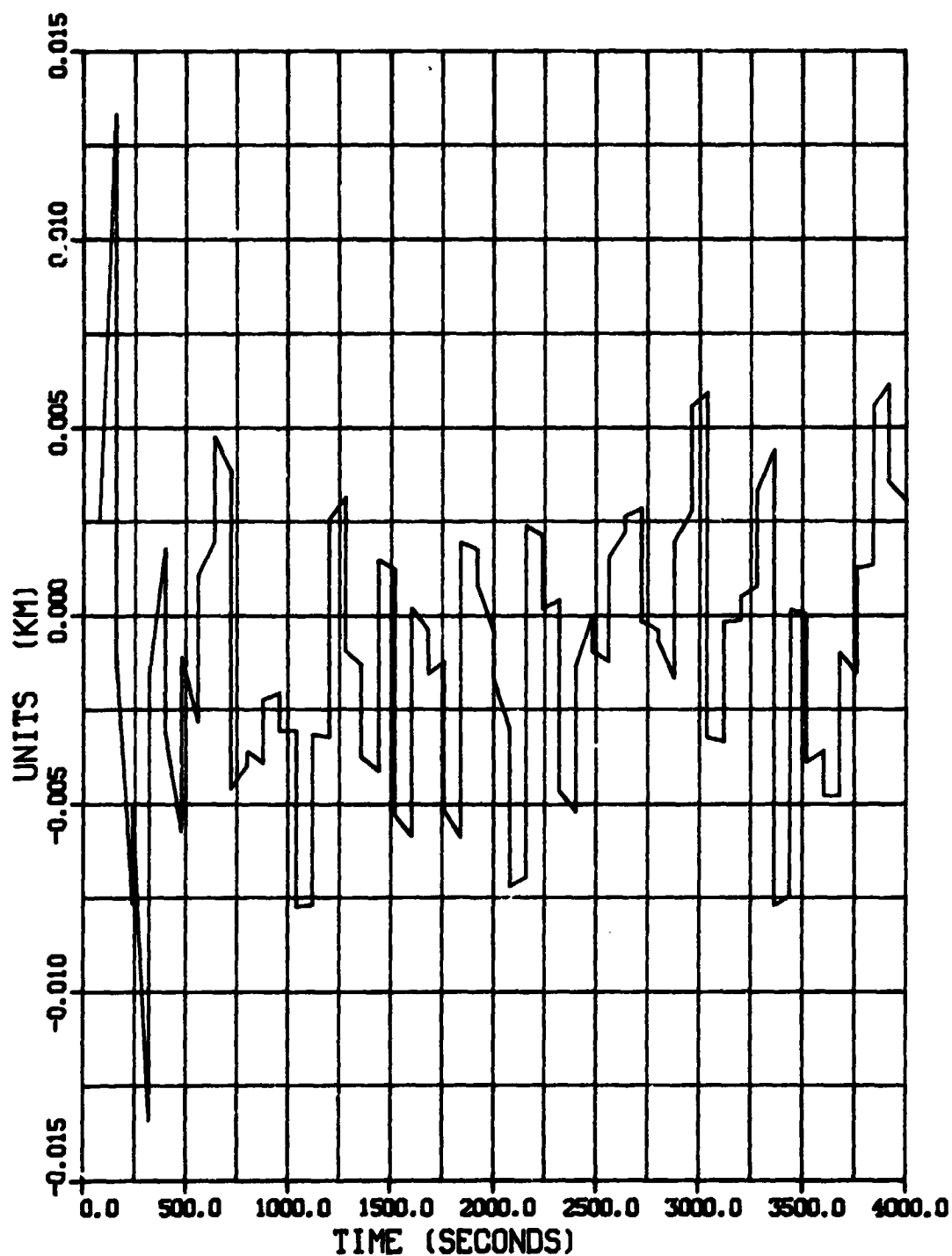
CASE23-DATA GCP-80SEC GPS-40SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

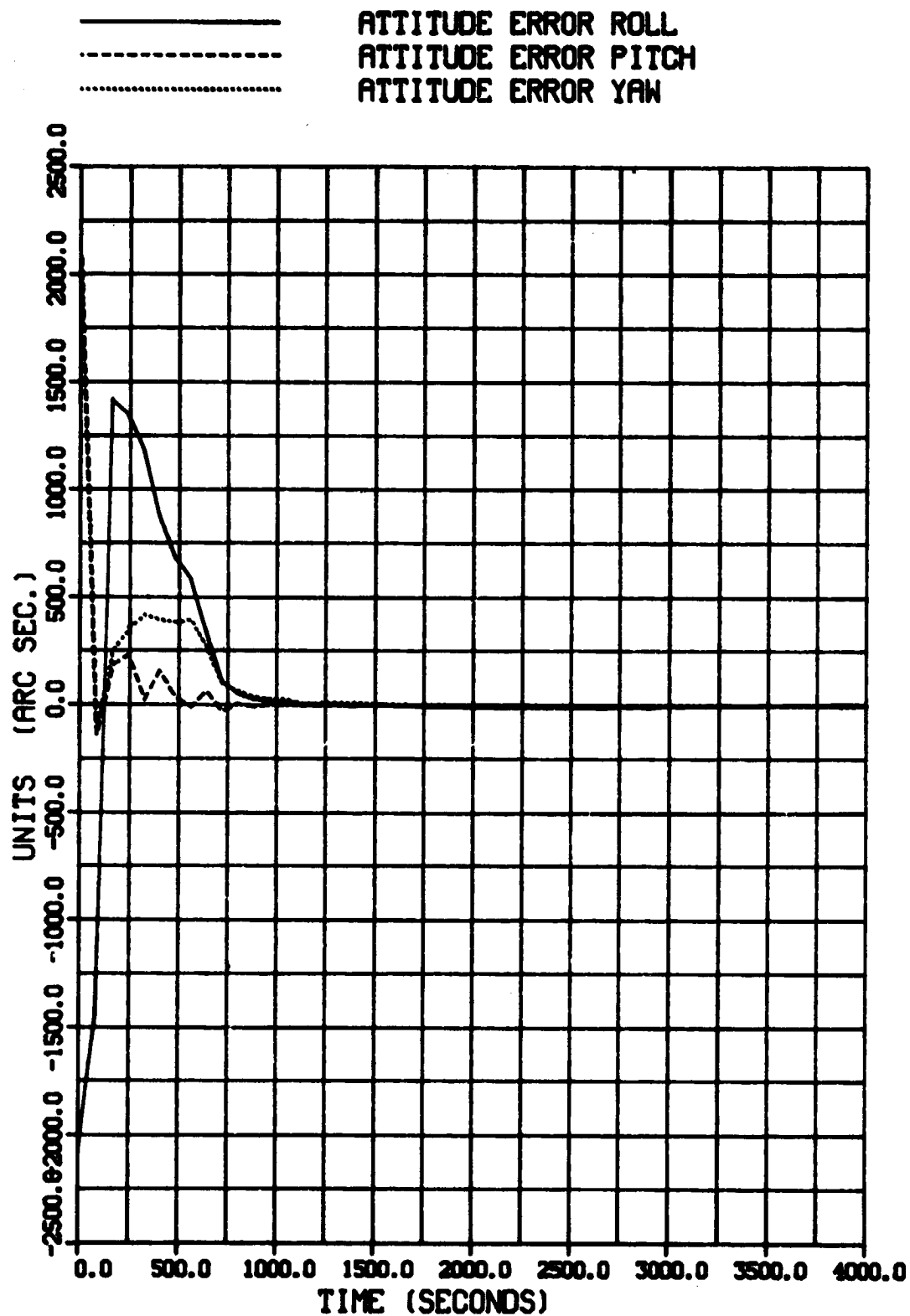


CASE24-DATA GCP-80SEC GPS-80SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

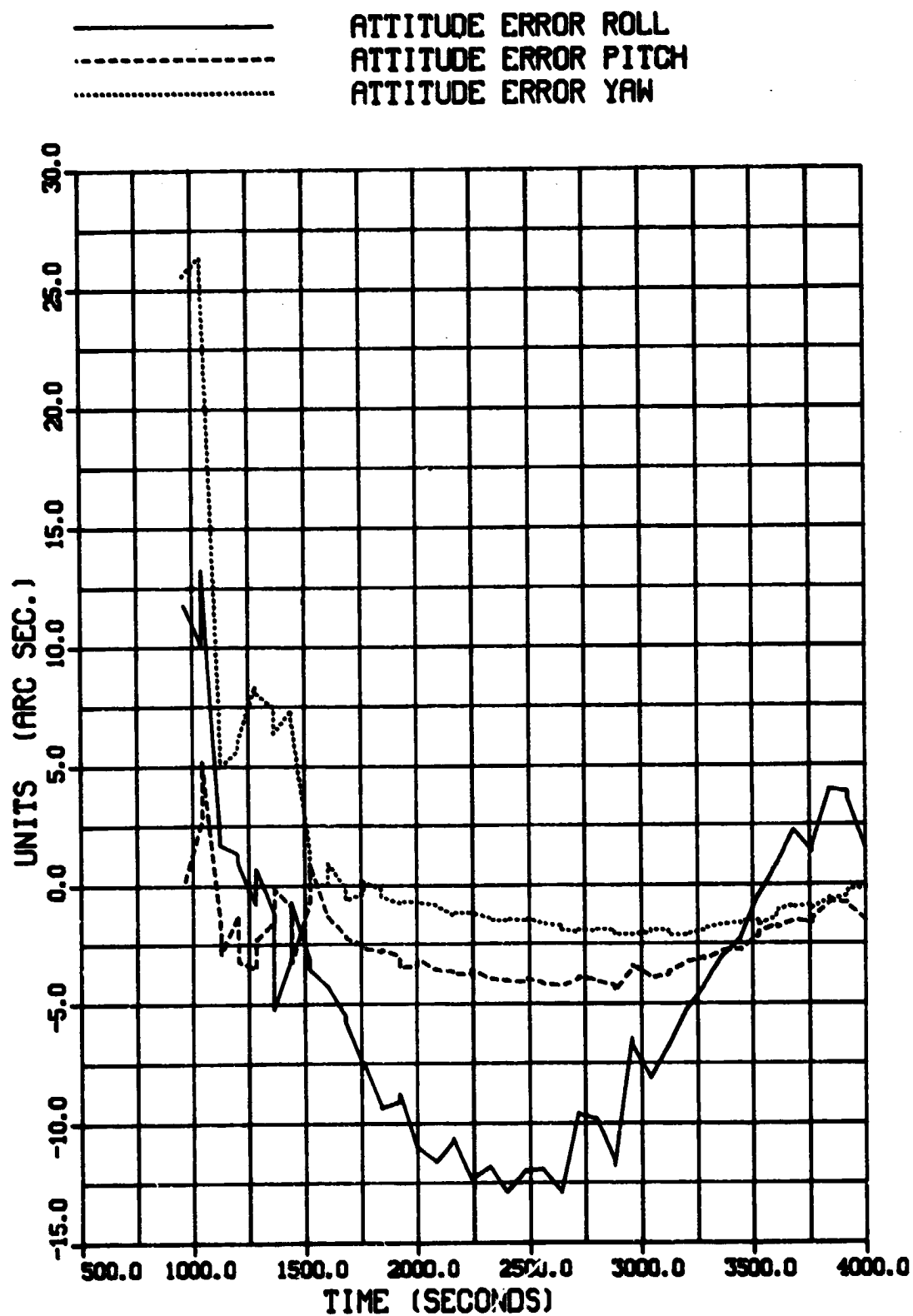
NAVIGATION X POSITION ERROR



CASE24-DATA GCP-80SEC GPS-80SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

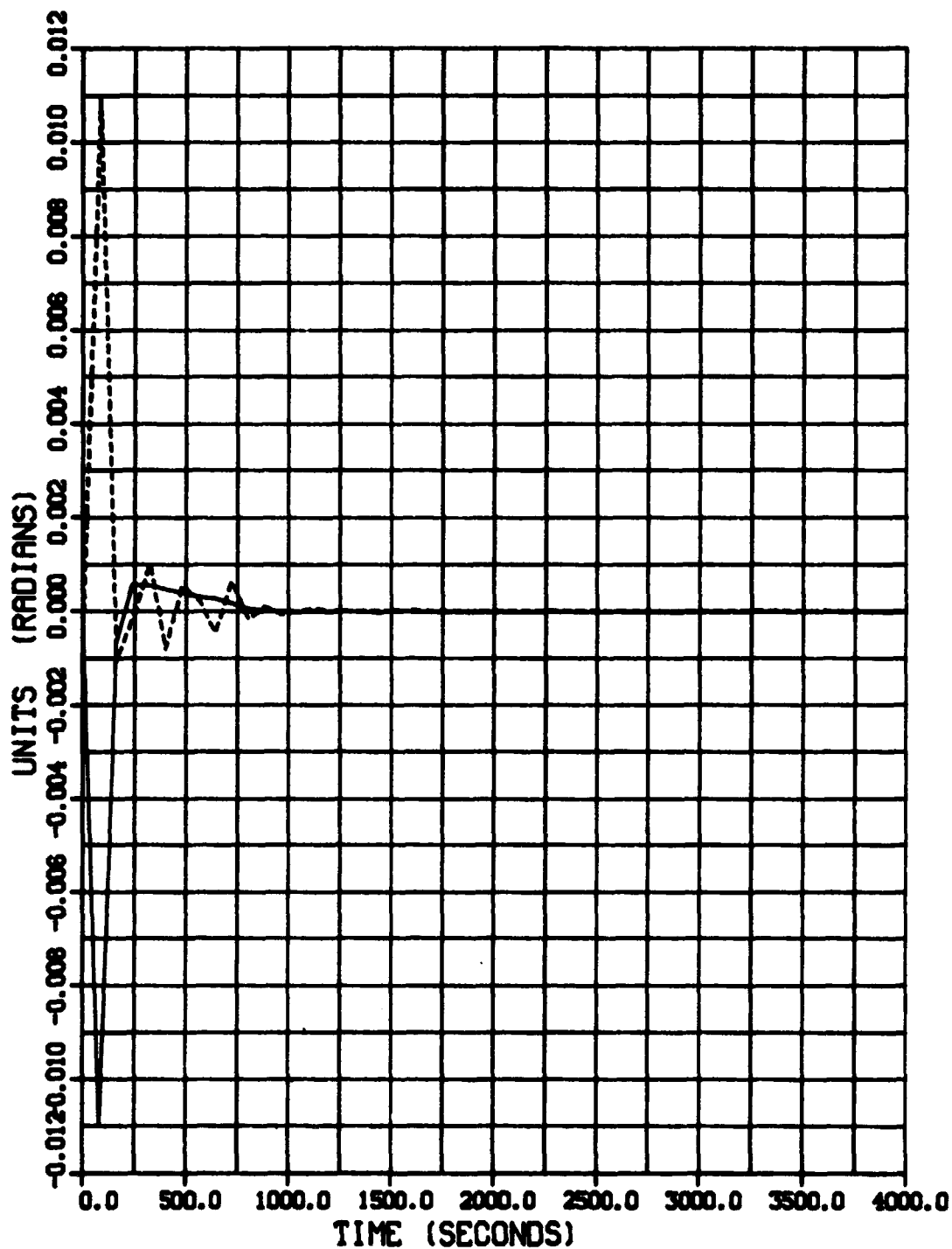


CASE24-DATA GCP-80SEC GPS-80SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***



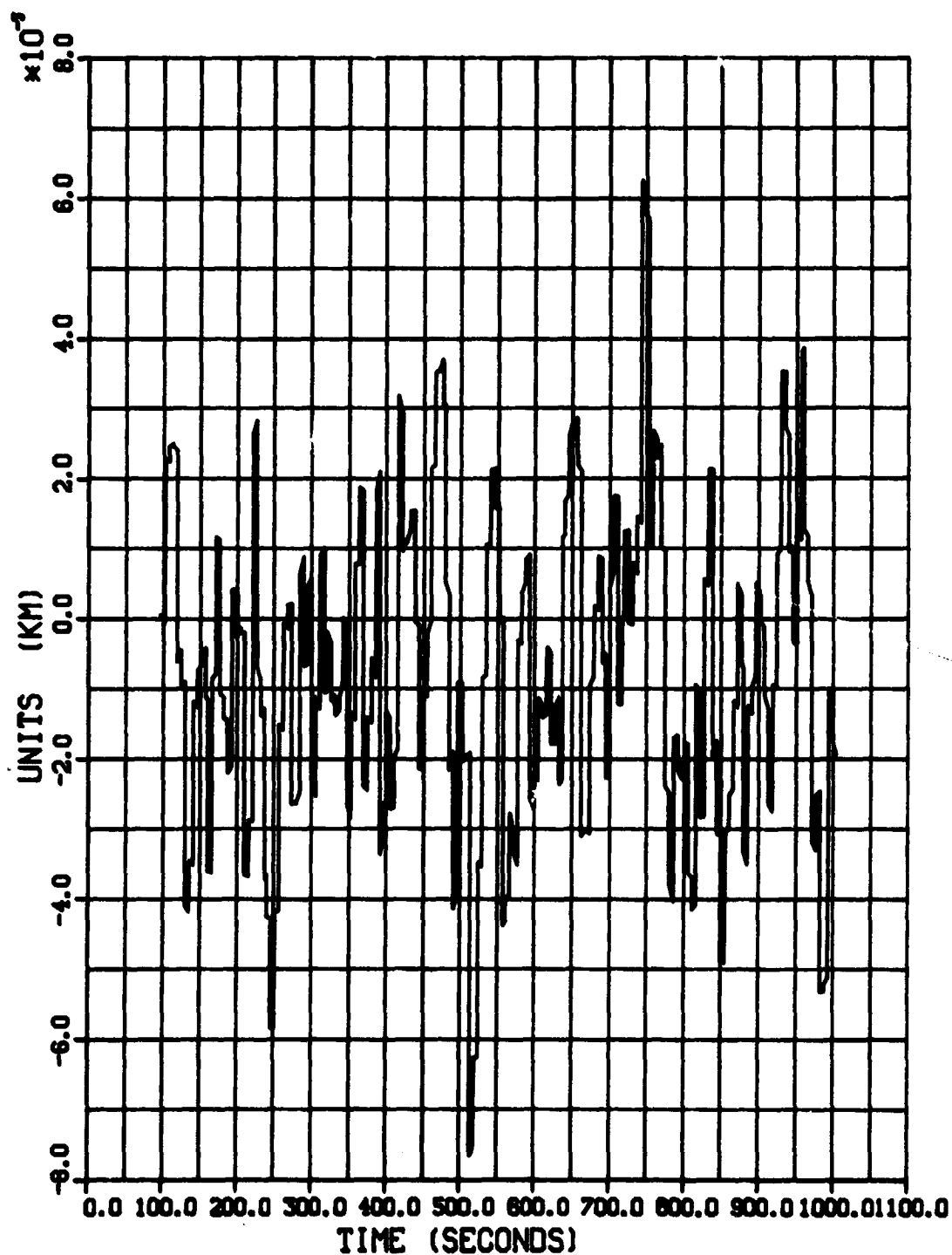
CASE24-DATA GCP-80SEC GPS-80SEC LA-RAND
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

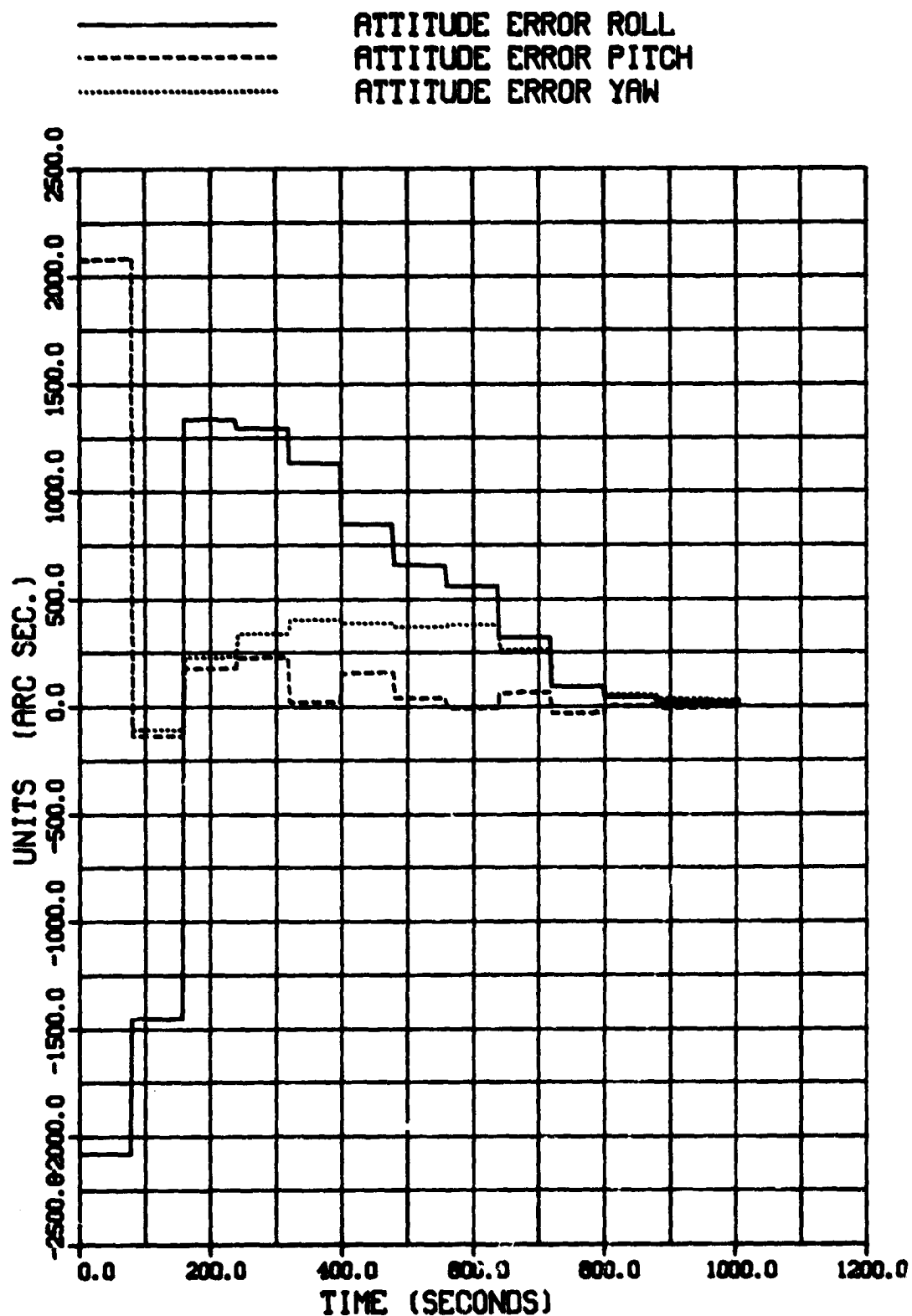


CASE25-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

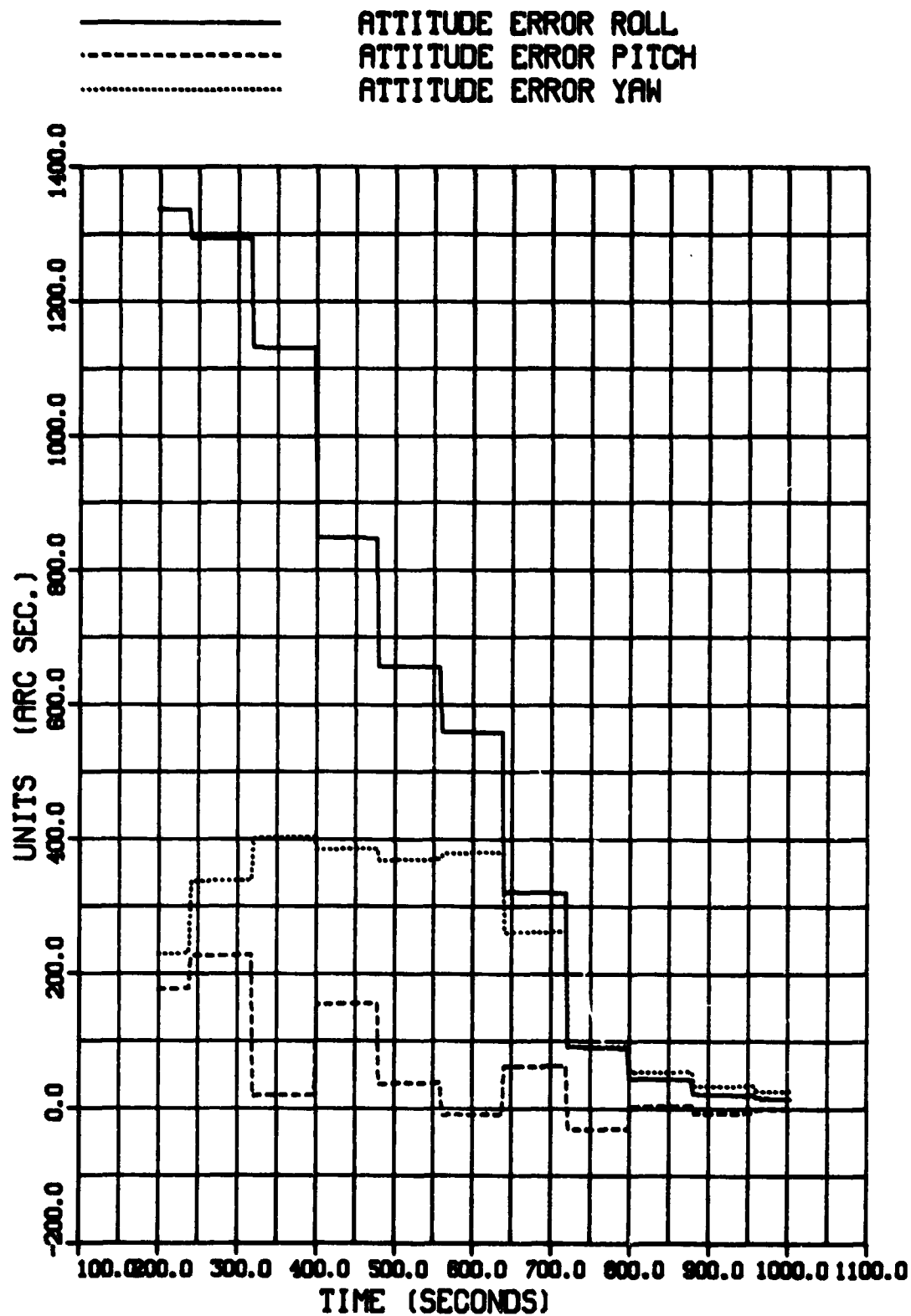
NAVIGATION X POSITION ERROR



CASE25-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

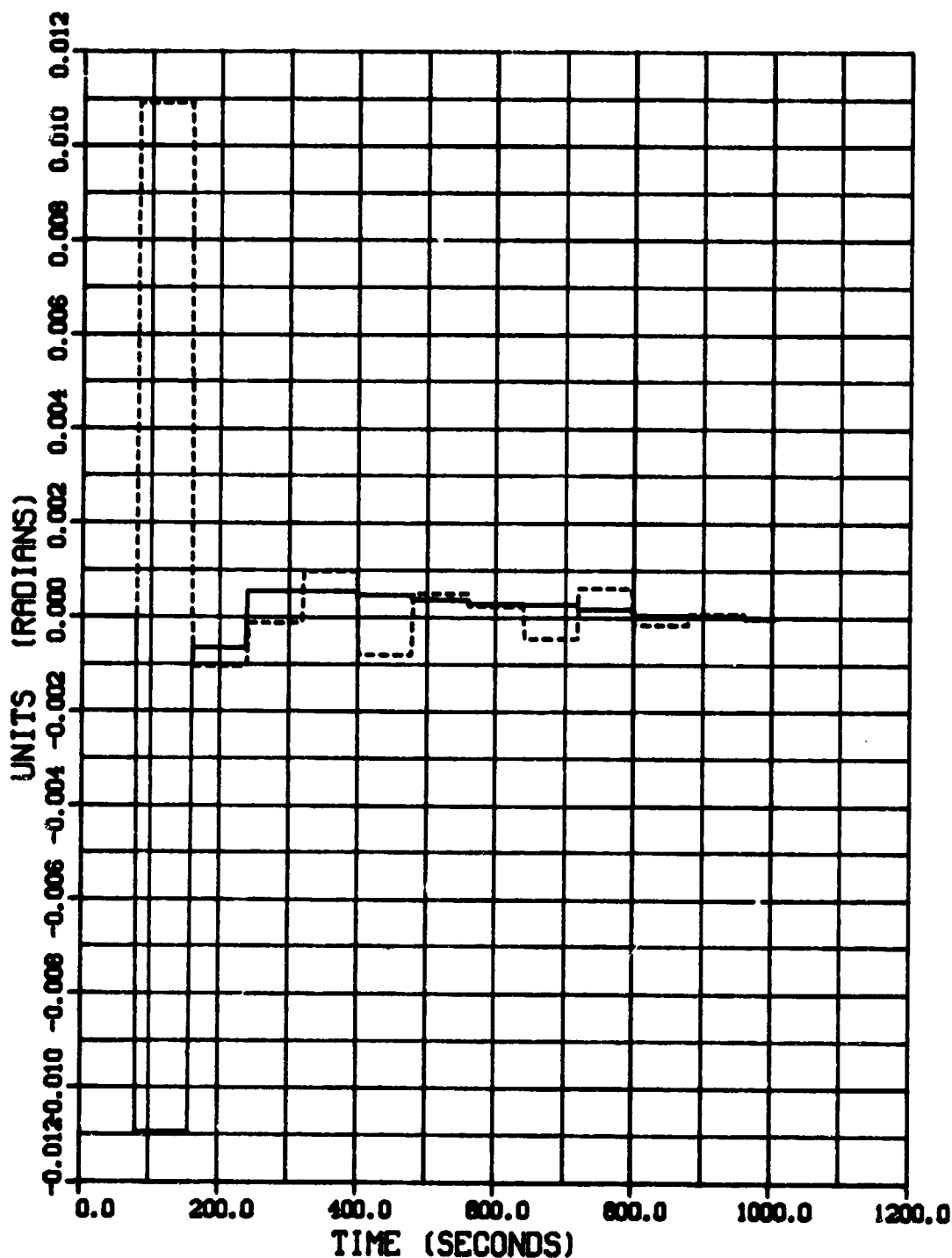


CASE25-DATA GCP-80SEC GPS-5SEC LA-RANDO
 *** INERTIAL COORDINATE FRAME ***



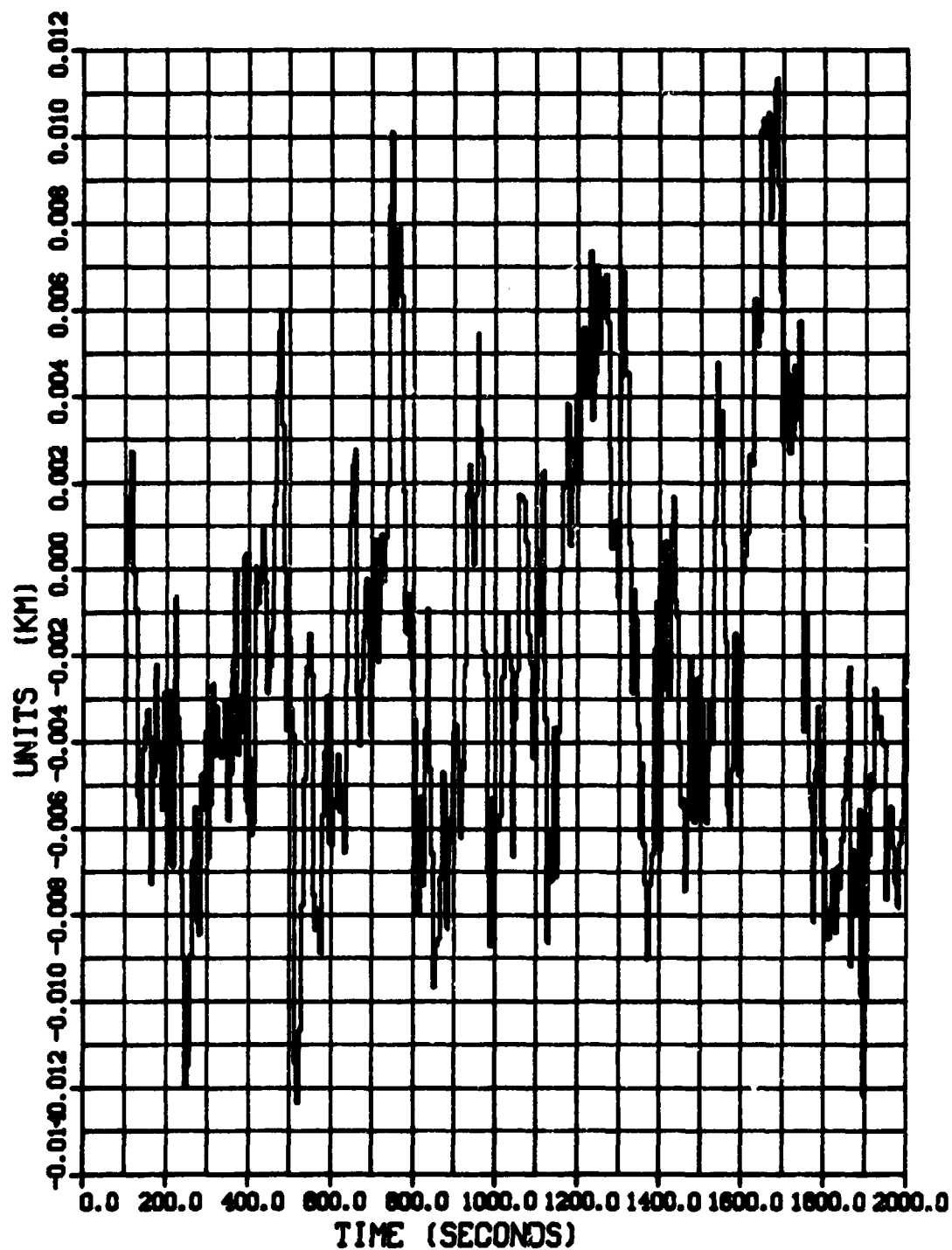
CASE25-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL



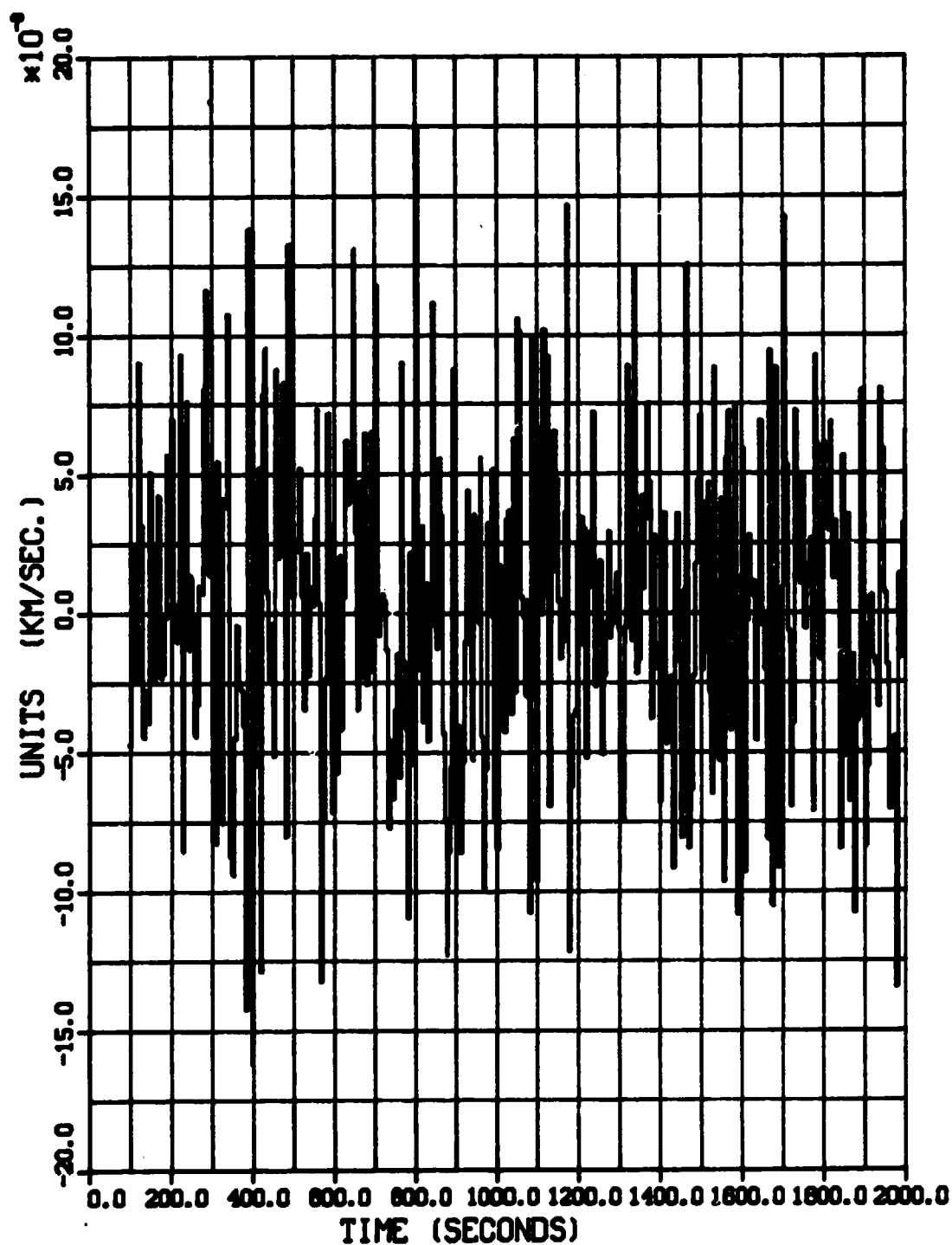
CASE26-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

NAVIGATION X POSITION ERROR

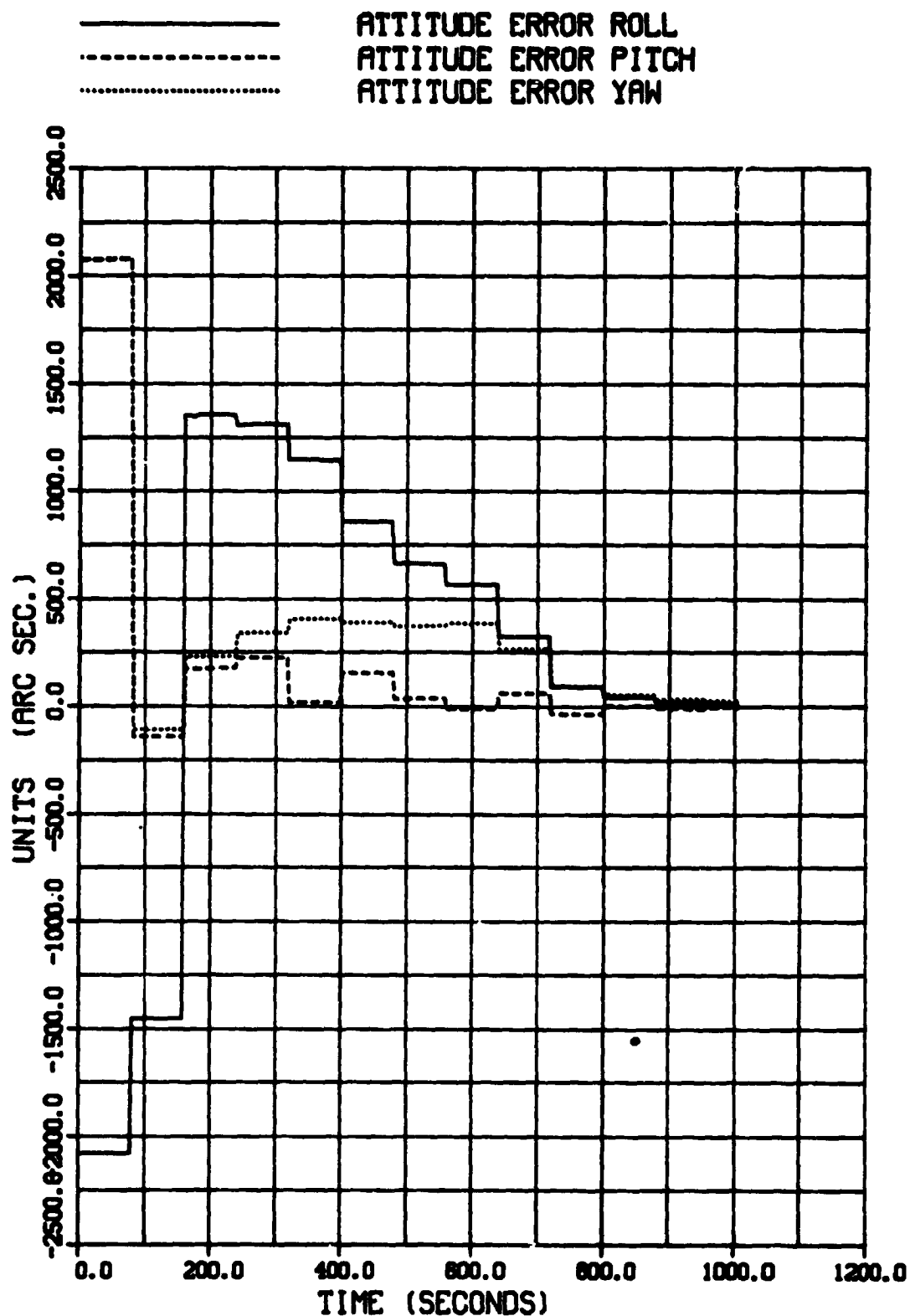


CASE26-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

NAVIGATION X VELOCITY ERROR

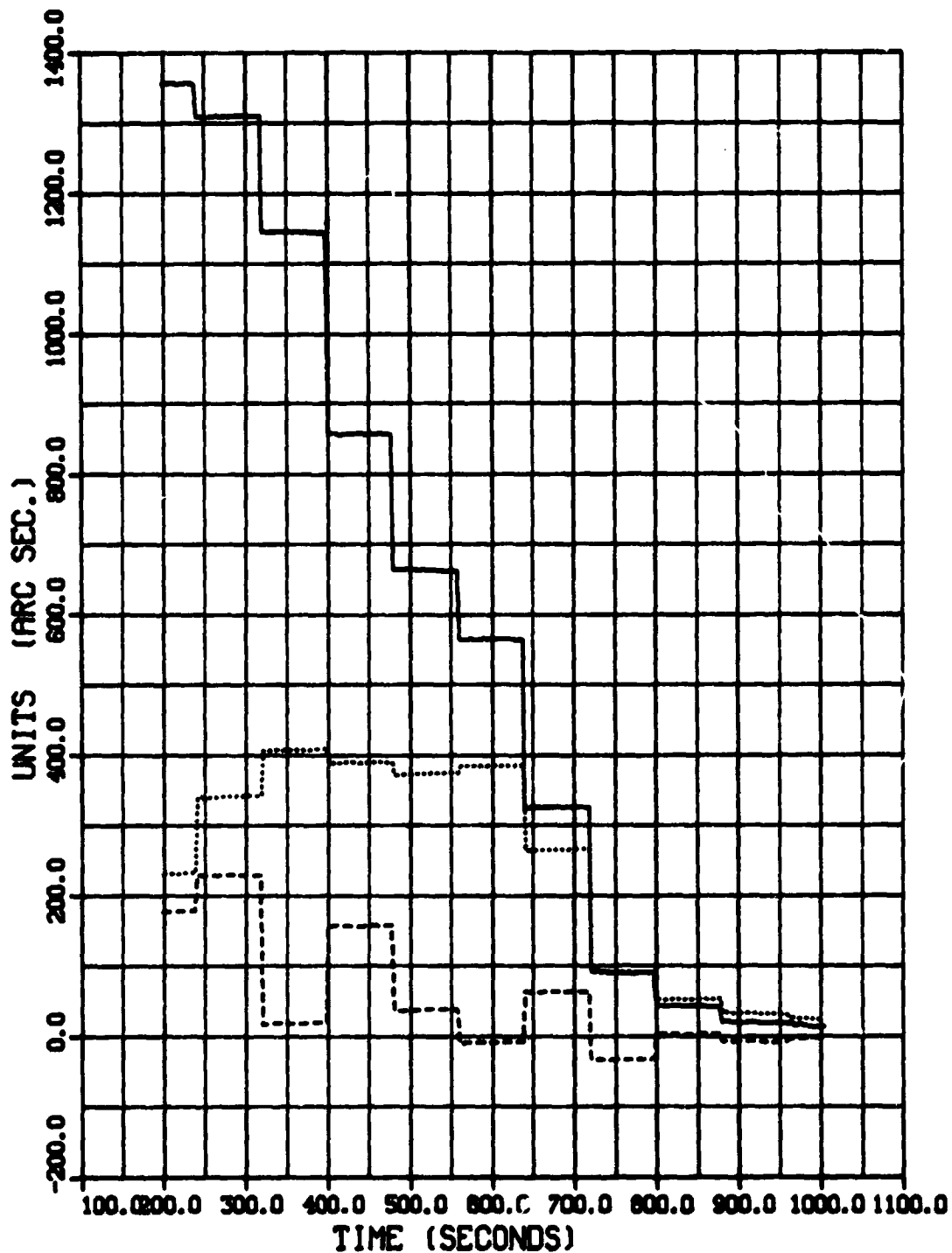


CASE26-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***



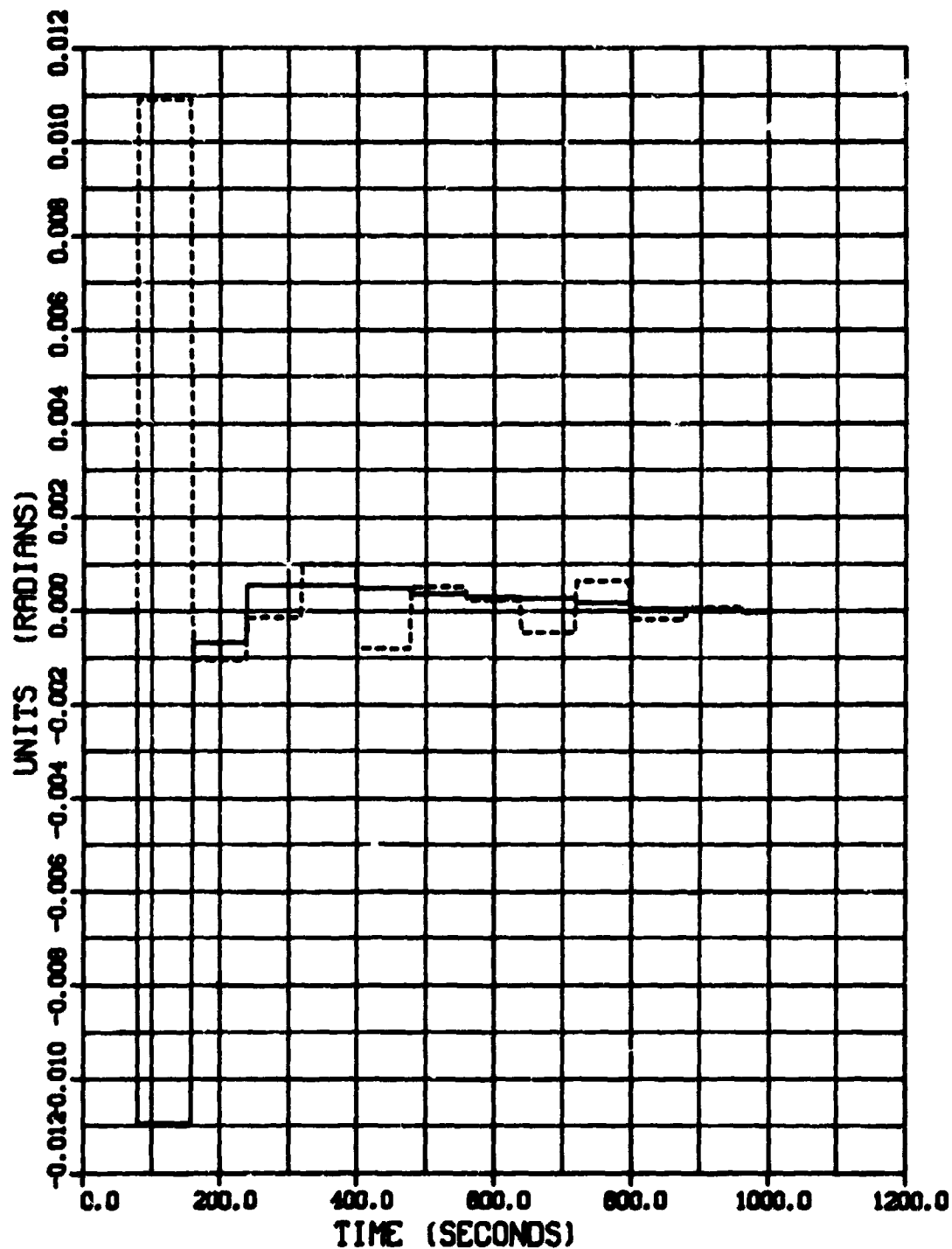
CASE26-DATA GCF-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

_____	ATTITUDE	ERROR	ROLL
-----	ATTITUDE	ERROR	PITCH
.....	ATTITUDE	ERROR	YAW



CASE26-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

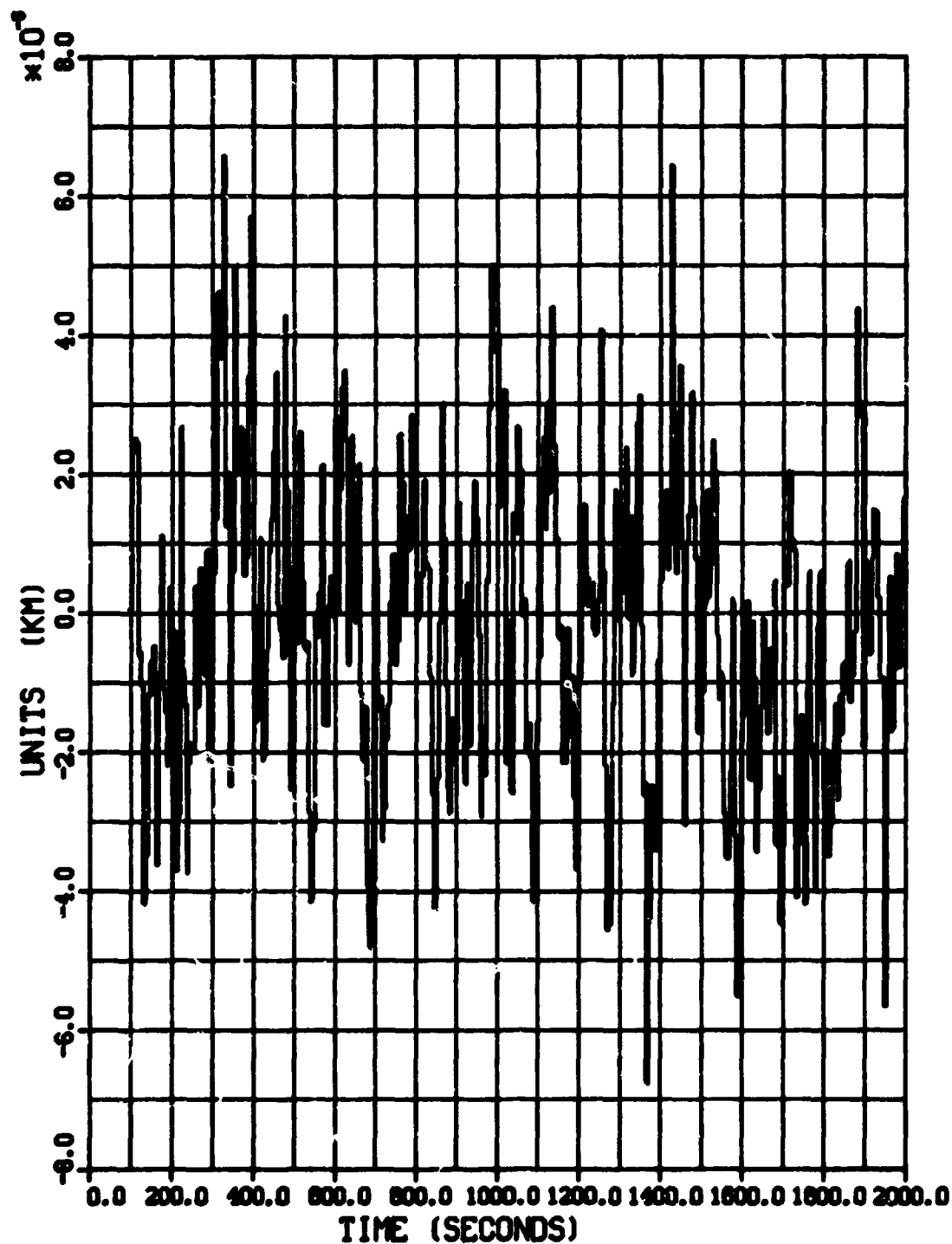
—— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL



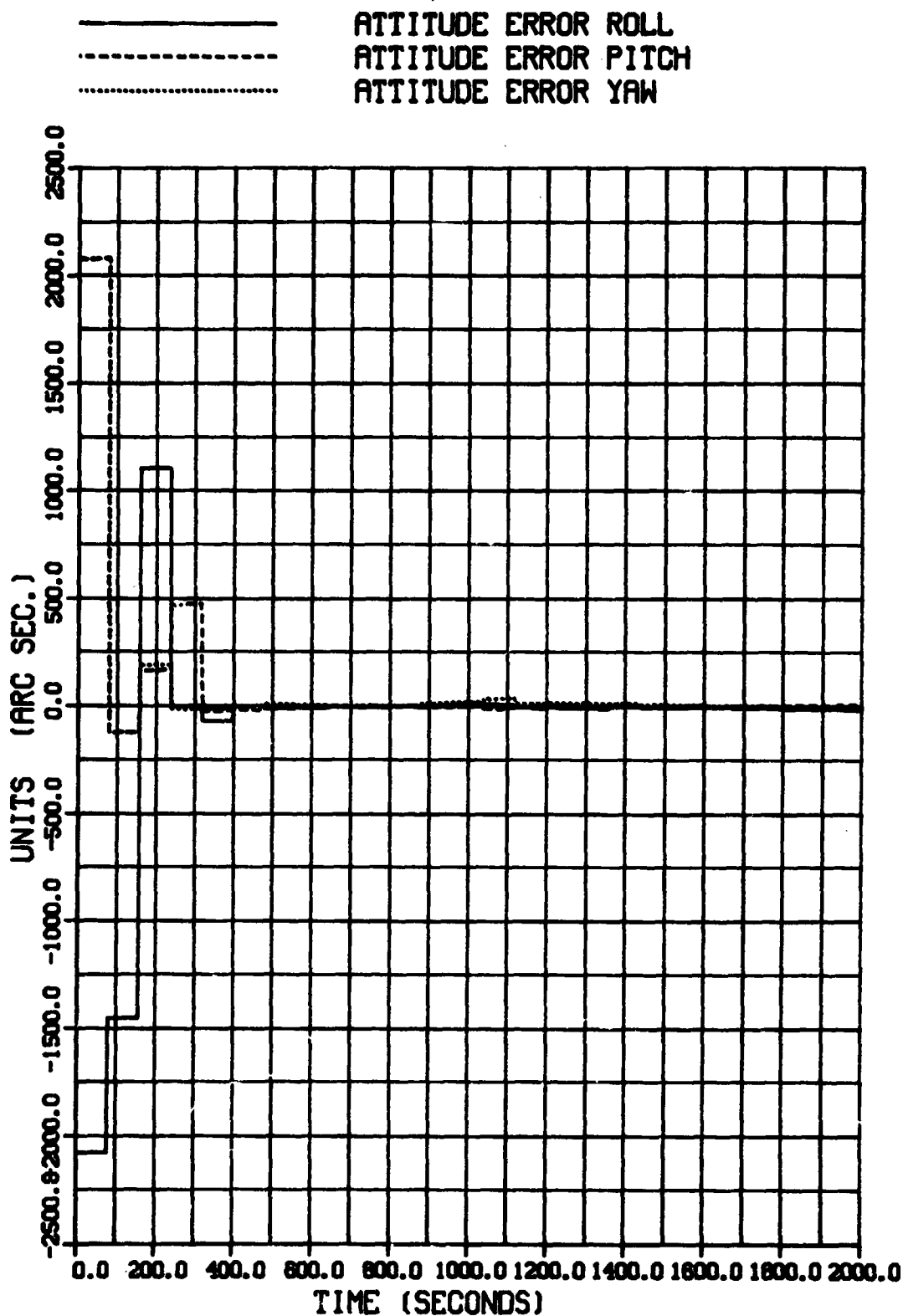
CASE30-DATA GCP-80SEC GPS-5SEC LA-RANDO

*** INERTIAL COORDINATE FRAME ***

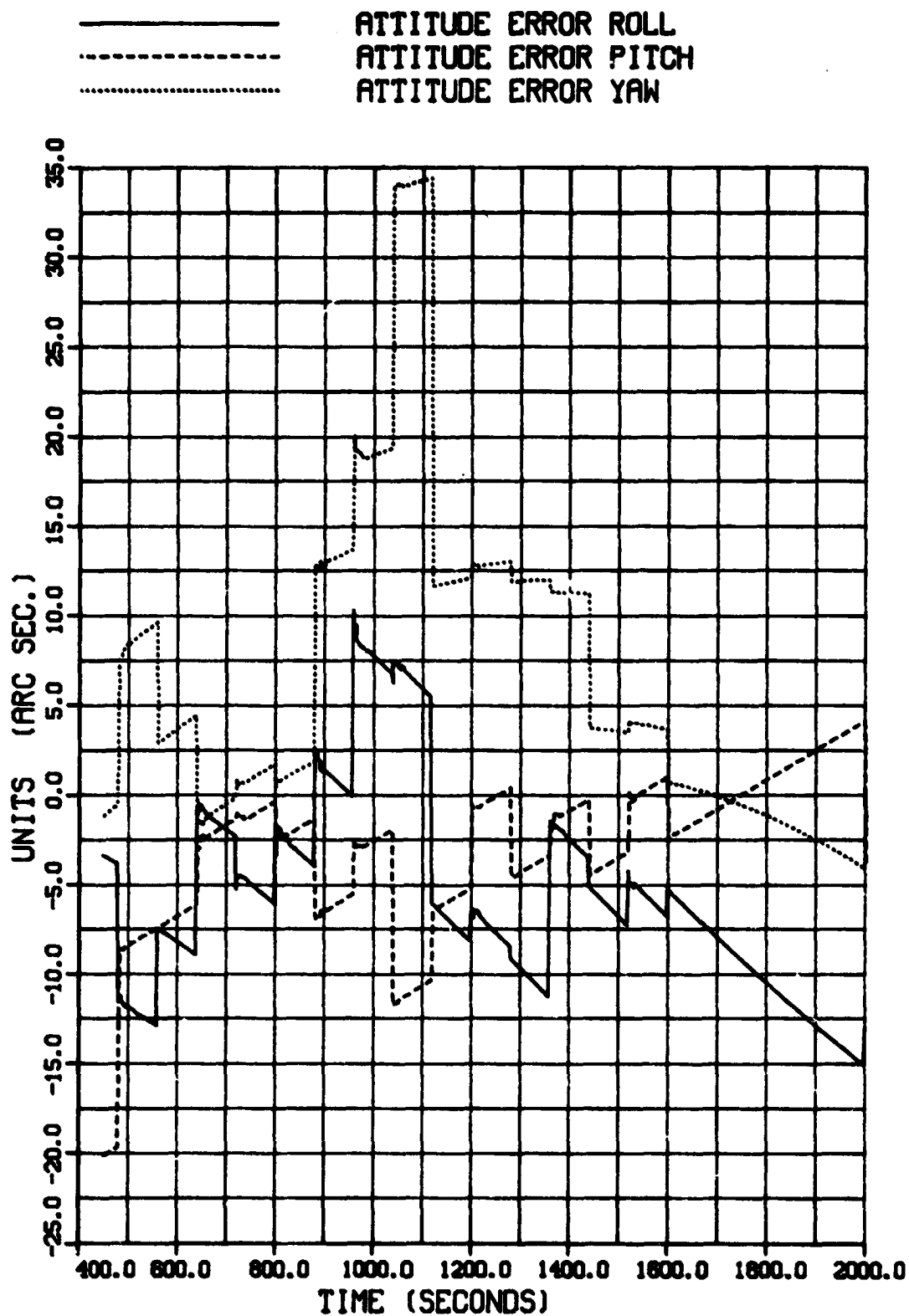
NAVIGATION X POSITION ERROR



CASE30-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

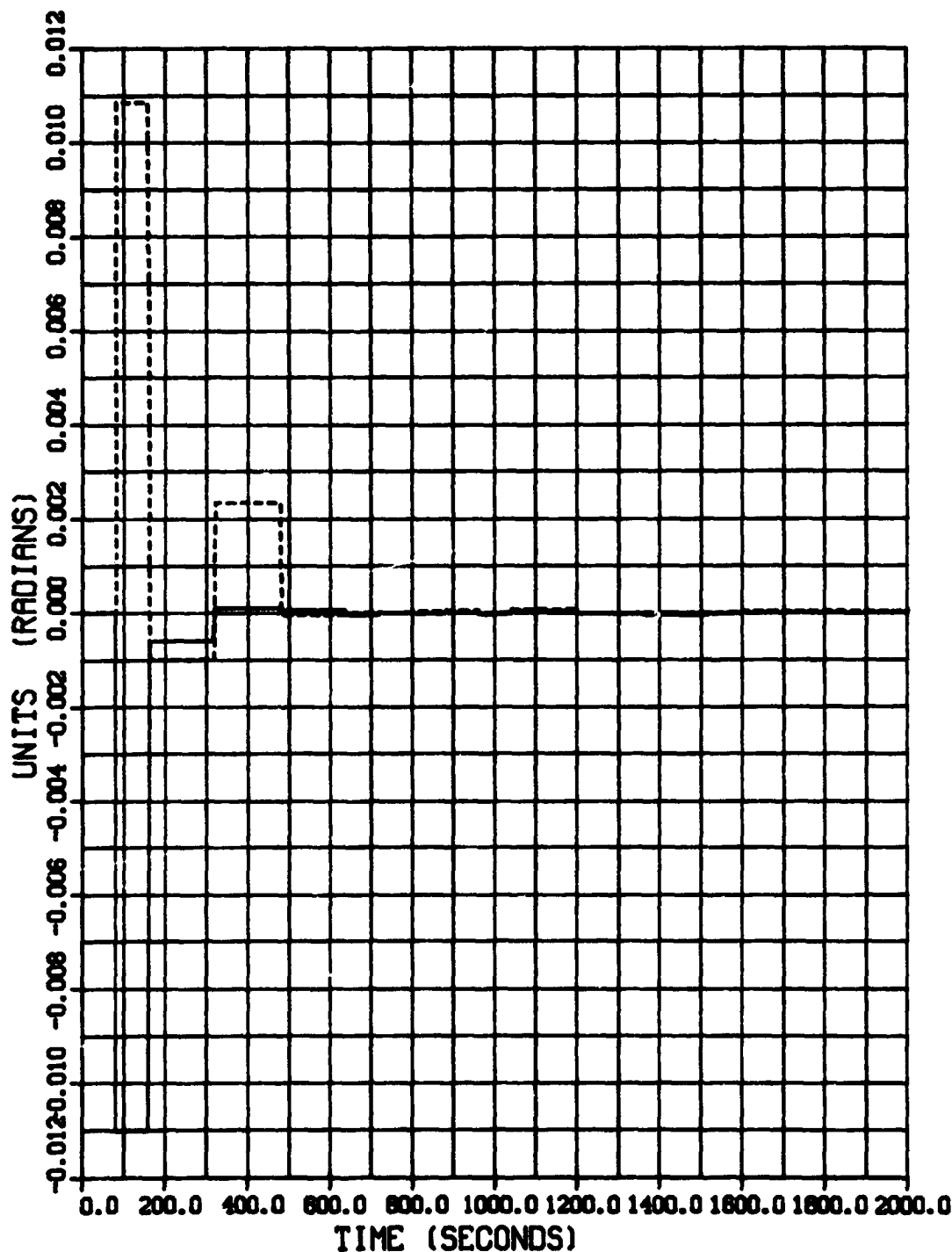


CASE30-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***



CASE30-DATA GCP-80SEC GPS-5SEC LA-RAND0
*** INERTIAL COORDINATE FRAME ***

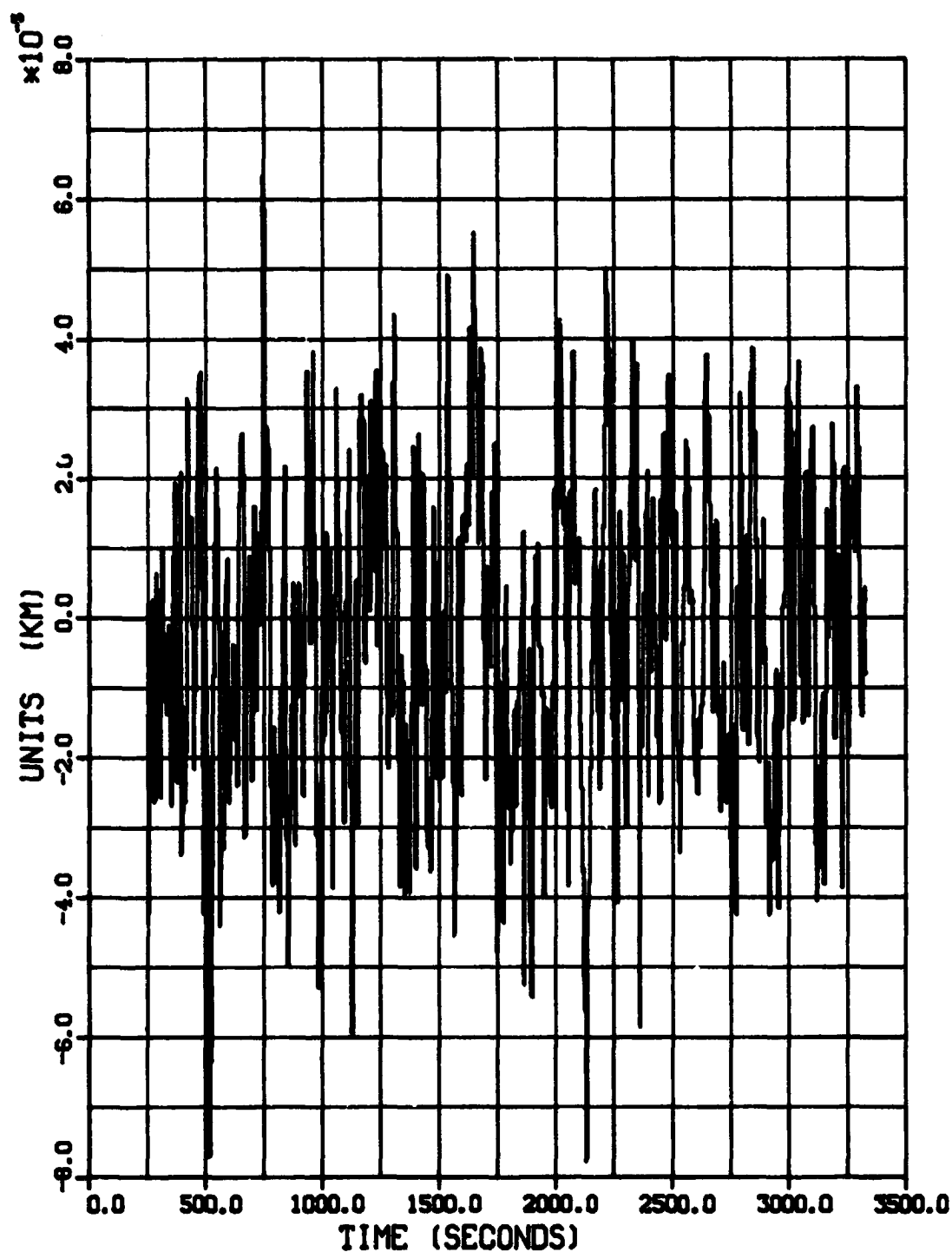
———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL



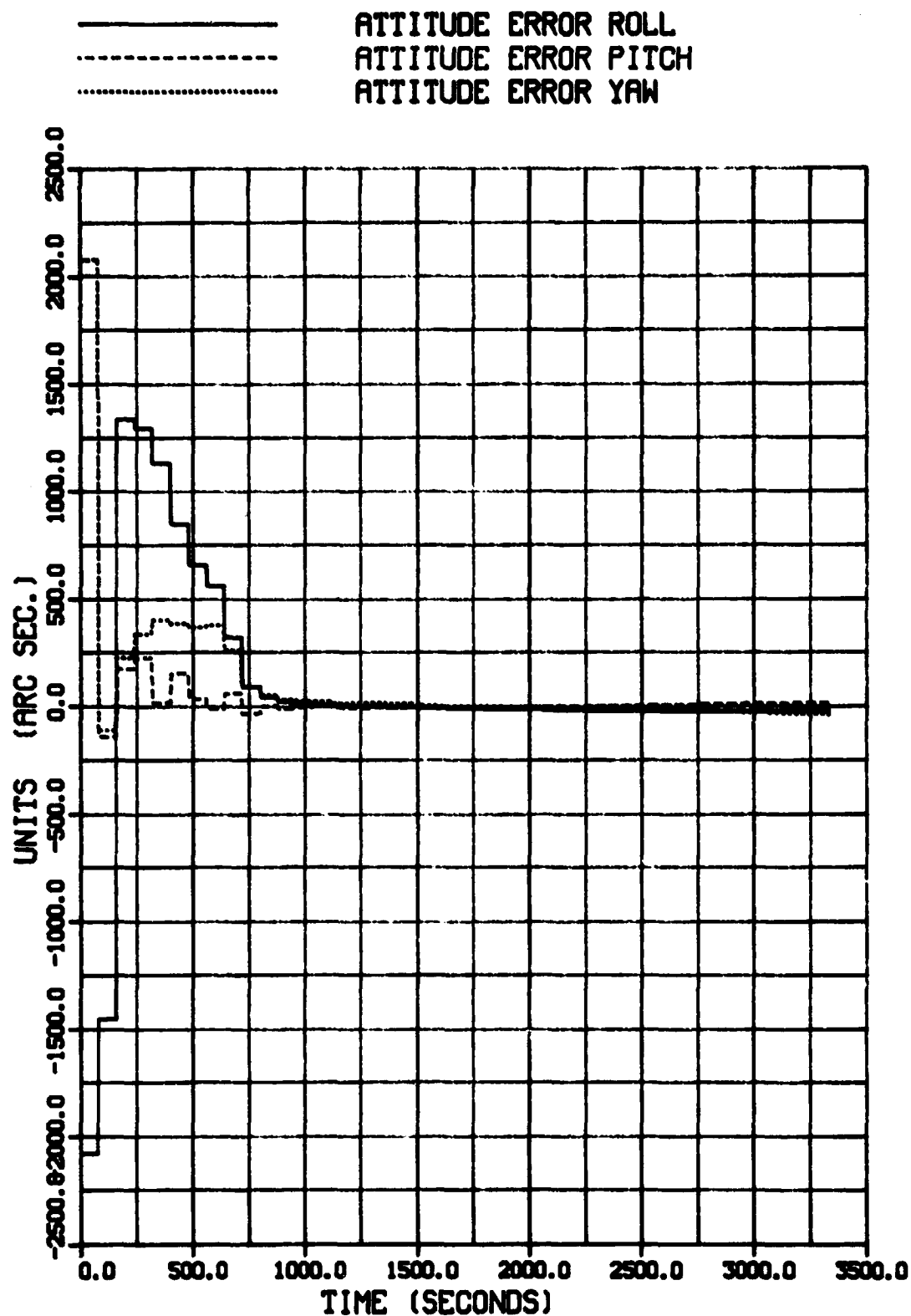
CASE31-DATA GCP-80SEC GPS-5SEC LA-RANDO

*** INERTIAL COORDINATE FRAME ***

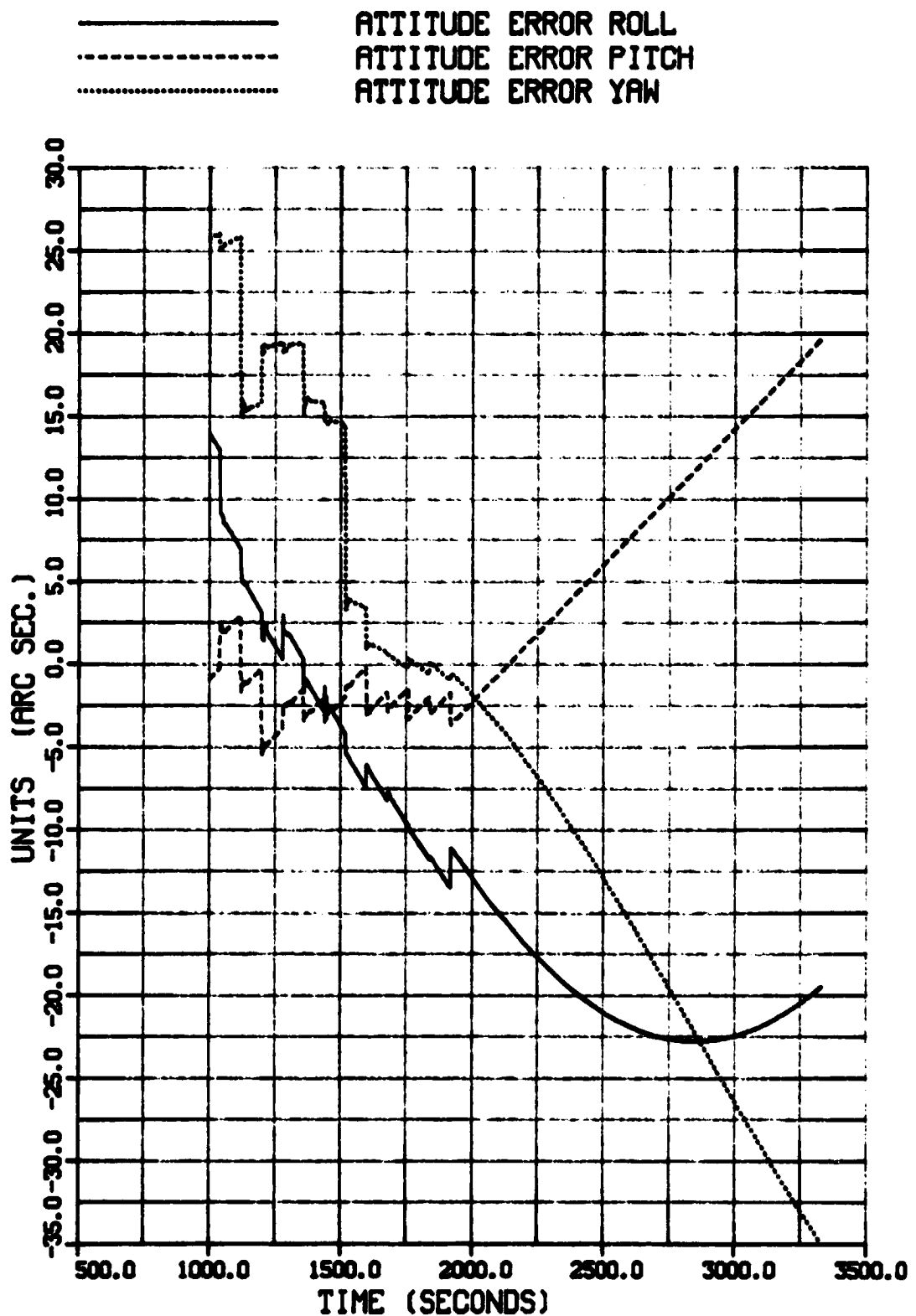
NAVIGATION X POSITION ERROR



CASE31-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***



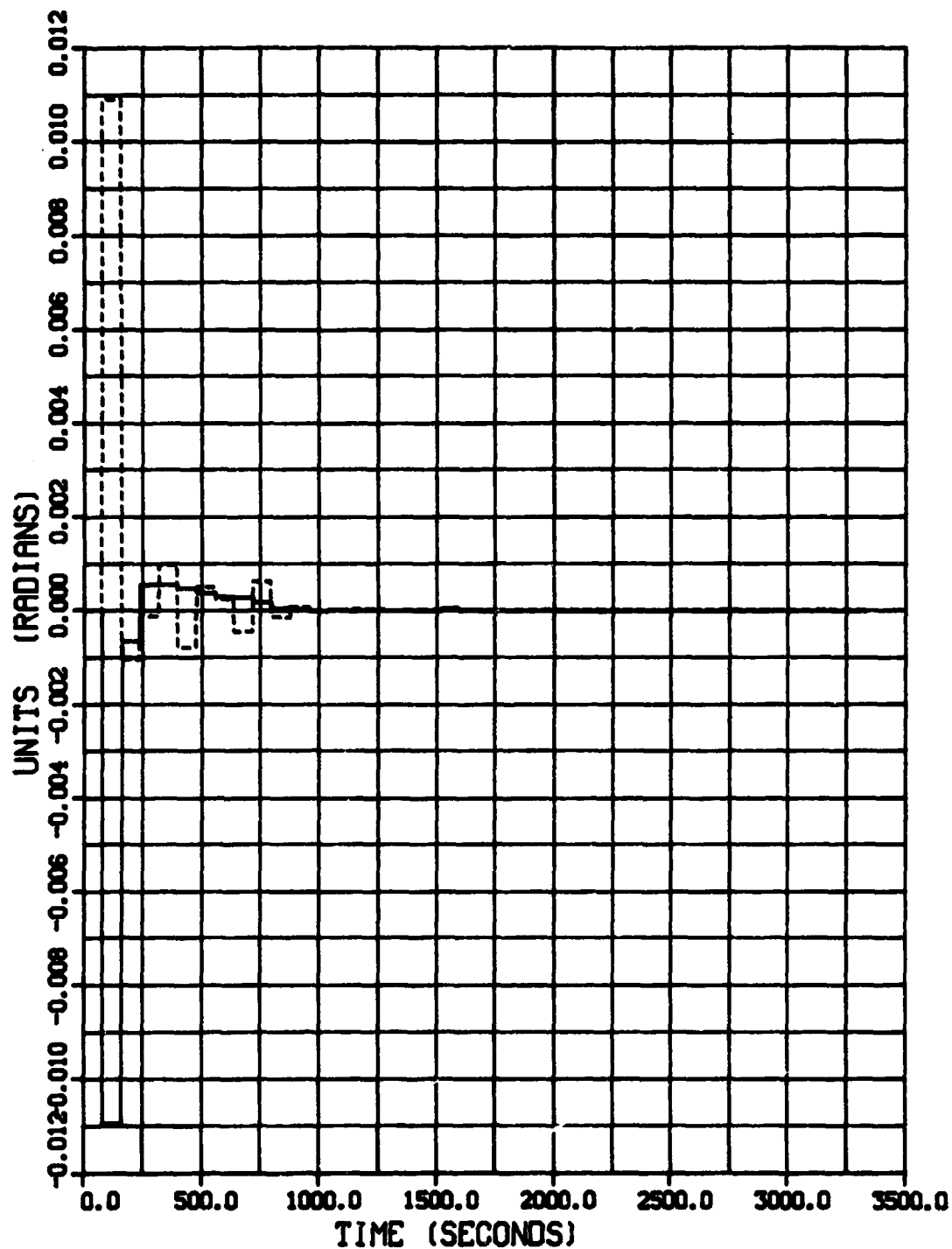
CASE31-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***



CASE31-DATA GCP-80SEC GPS-5SEC LA-RANDO

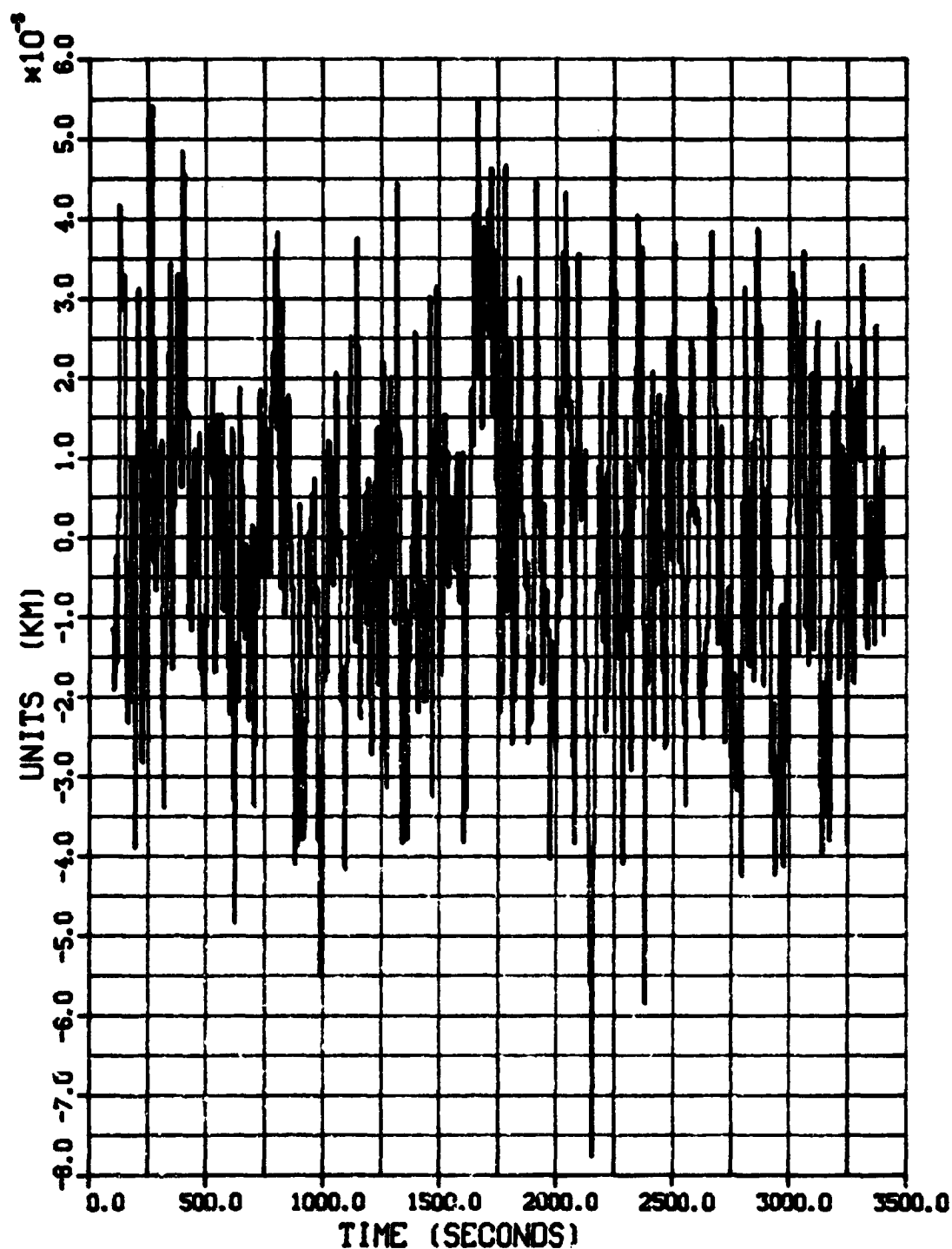
*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL

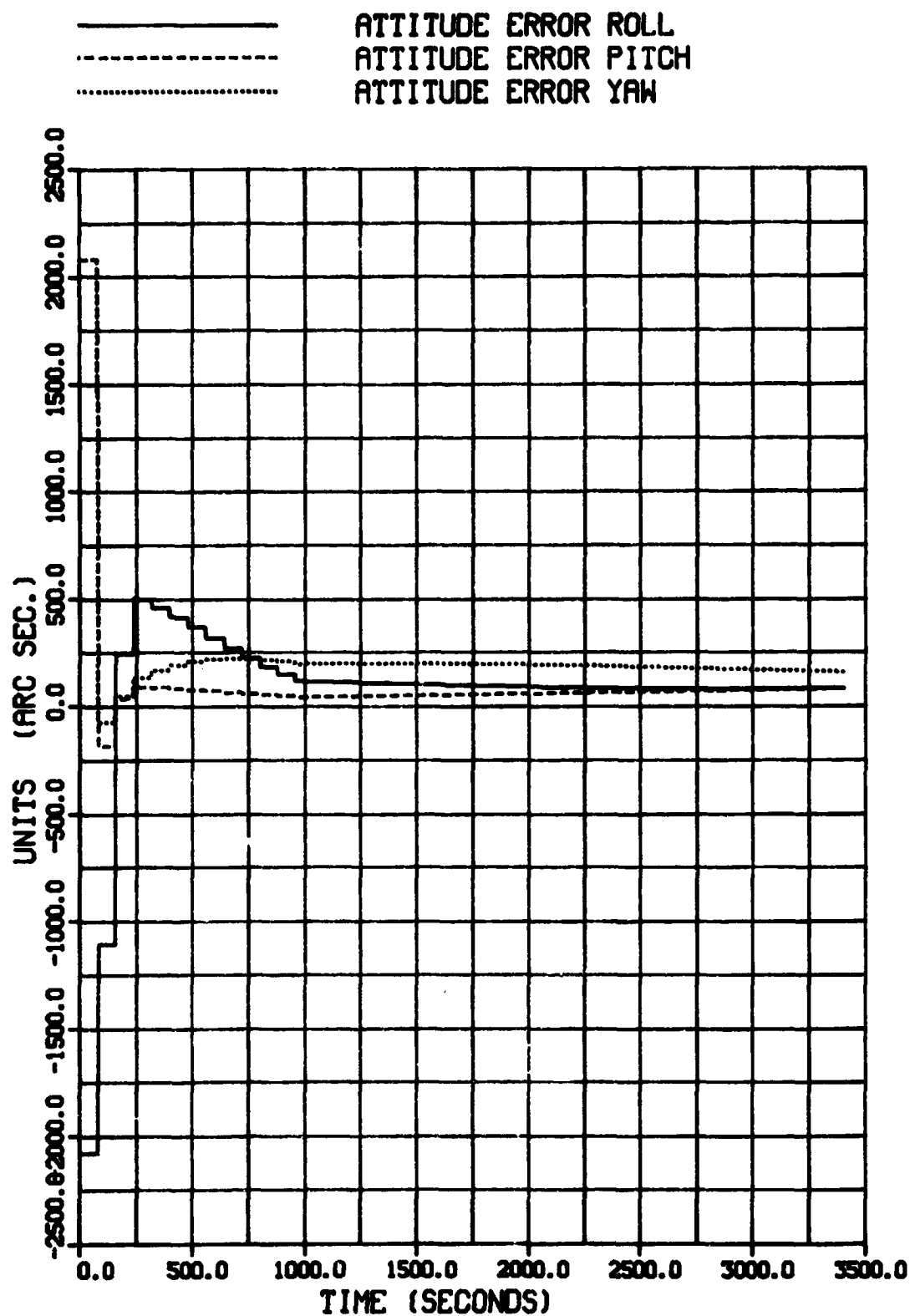


CASE32-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***

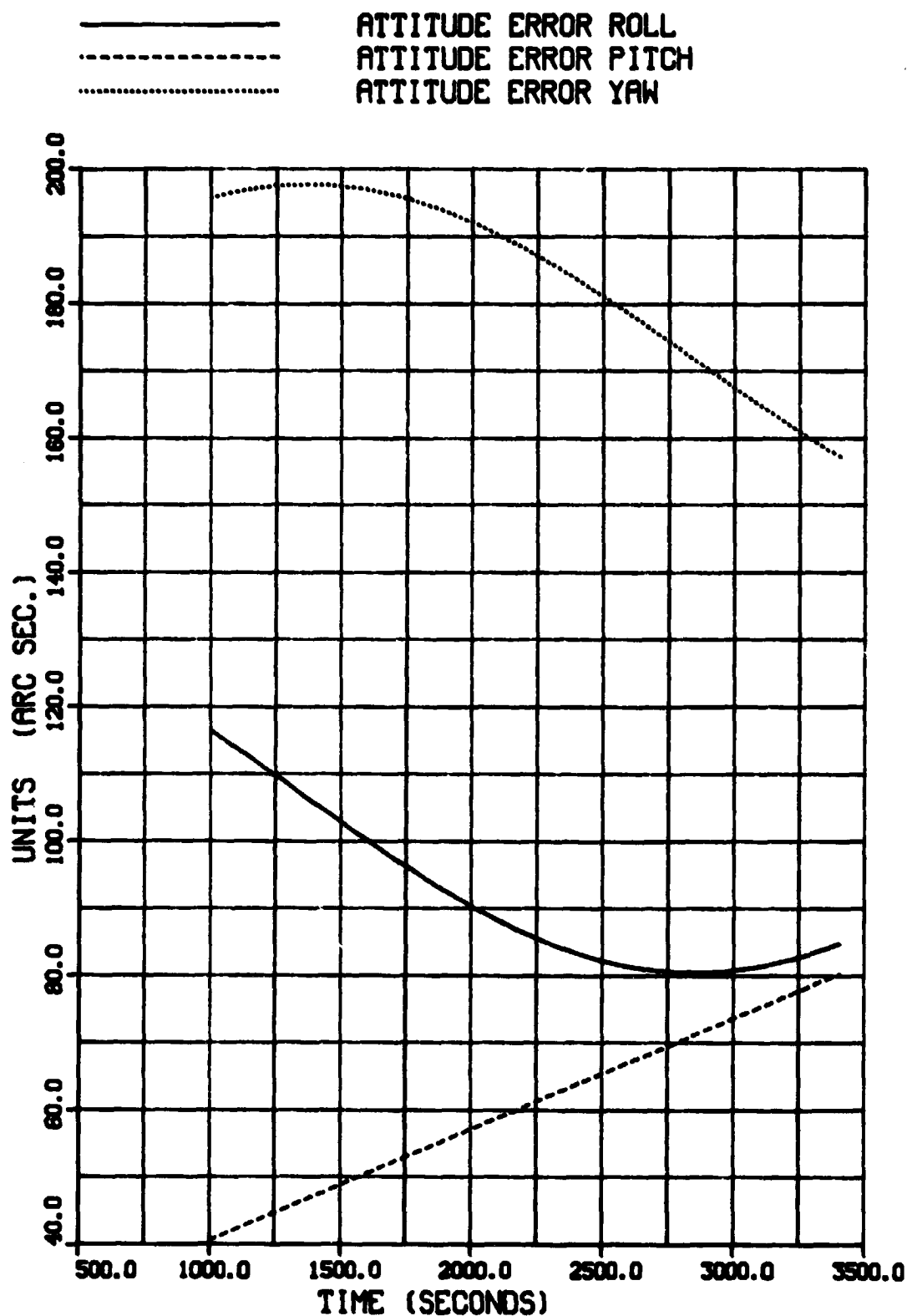
NAVIGATION X POSITION ERROR



CASE32-DATA GCP-80SEC GPS-5SEC LA-RANDO
*** INERTIAL COORDINATE FRAME ***



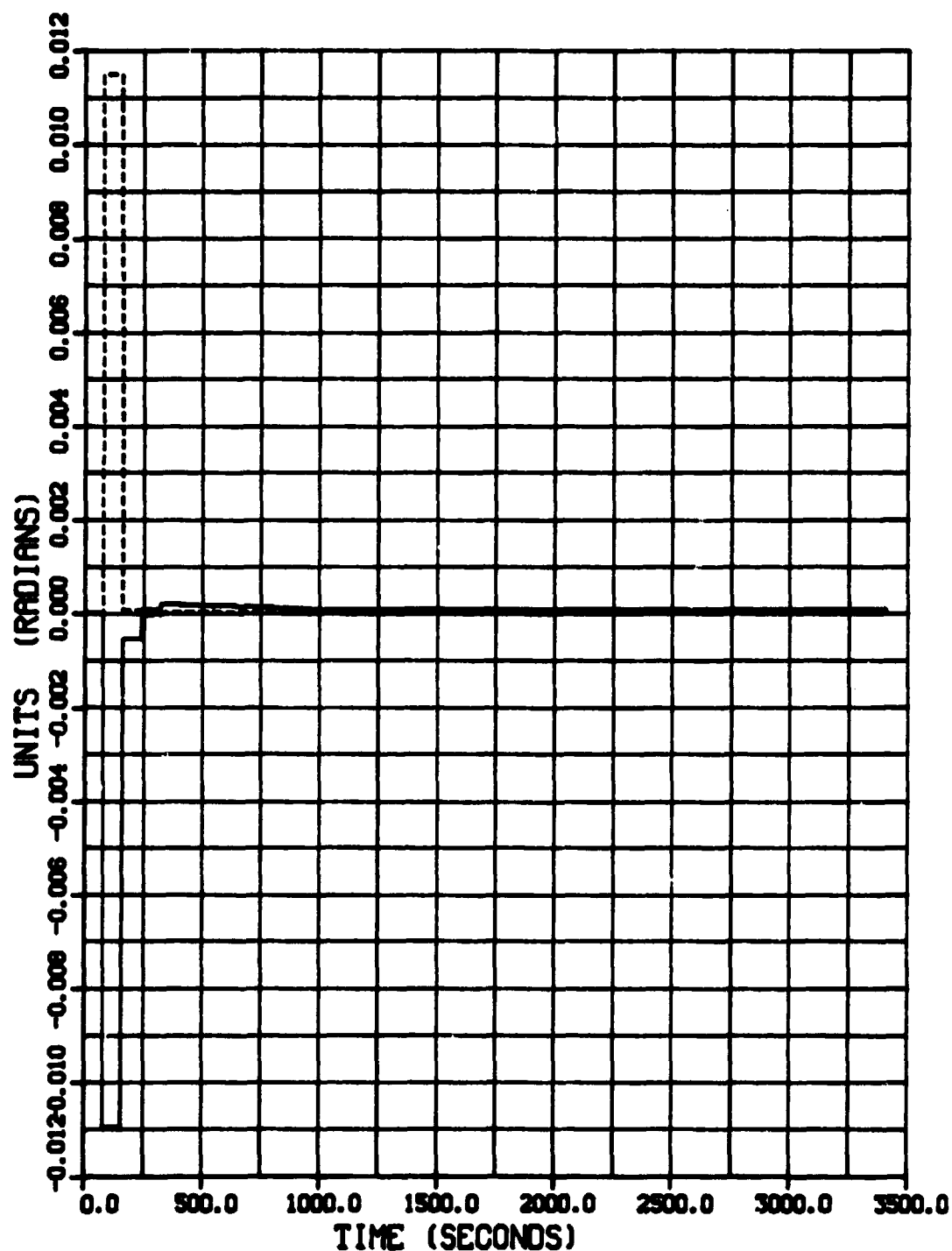
CASE32-DATA GCP-80SEC GPS-5SEC LA-RANDO
 *** INERTIAL COORDINATE FRAME ***



CASE32-DATA GCP-80SEC GPS-5SEC LA-RANDO

*** INERTIAL COORDINATE FRAME ***

———— L.M. TRACKER RESIDUAL HORIZONTAL
----- L.M. TRACKER RESIDUAL VERTICAL



V. IMAGE CORRECTION ACCURACY MEASUREMENT

The images generated by the Landsat multispectral scanner (MSS) suffer from a variety of geometric distortions. Users of Landsat image products require that the temporal registrations between scenes remain less than half a pixel. Registration of images requires the measurement of displacements through the use of GCP detection, calculation of geometric distortion coefficients, and resampling of the raw data to provide imagery with uniform pixel spacing. If any of these functions are in error, the final image will have some registration error. There are no procedures for determining the quality of registration provided by the Goddard Space Flight Center. Such a technique must be developed to periodically measure the performance of the image correction system so models that compute the distortion coefficients may be updated as the MSS and satellite parameters vary. Under this contract a program was designed to perform the function of performance measurement.

The primary element in the image registration quality measurement technique had been previously developed by Martin Marietta. Although the programs were developed to analyze the accuracy of various correlation algorithms, a secondary output is the offset between the two images at each ground control point (GCP) location. A detailed description of these programs has been included in Volume III.

A great deal of insight into the image registration error sources may be gained by plotting a grid pattern that illustrates the offset between two images. Although this technique will not detect systematic registration errors, it provides a tool to study the random or periodic errors.

When examining the registration quality of a number of scenes, it is necessary to establish a reference image to which all others may be compared. To create the distortion grid, it must be assumed that the reference scene is a uniformly sampled orthogonal image (note that this assumption precludes the measurement of the registration accuracy in the reference scene but will provide for a relative accuracy measurement of the other scenes analyzed). The reference image may then be separated into a grid pattern (Fig. V-1) where each of the lines are uniformly spaced and orthogonal.

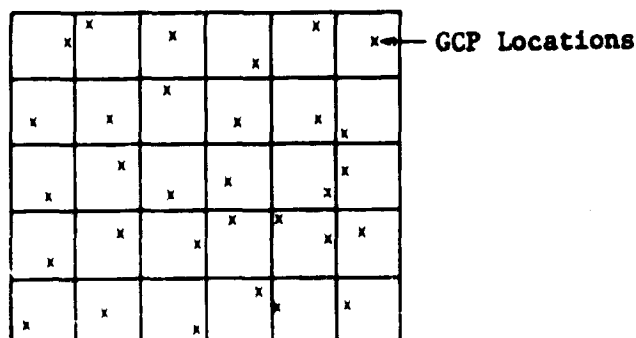


Figure V-1
Reference Grid Pattern

The distortion of the scenes may be measured by registering the imagery at each of the grid crossings in the reference with the imagery in the scenes being studied. The registration vector, which is then just a description of the relative offsets, provides a measurement of the local distortions. It is important to realize that a recognizable GCP will not be present at each grid crossing. Therefore GCPs may be selected from anywhere within the subarea. The correlation analysis program has been modified to allow display of a uniform orthogonal grid pattern over the reference scene and allow the selection of a GCP in each subarea in the grid. The spacing of the grid pattern will nominally be 200 pixels so 400 GCP registrations will be ideally provided. A visual control logic representation (VCLR) of this process is shown in Figure V-2.

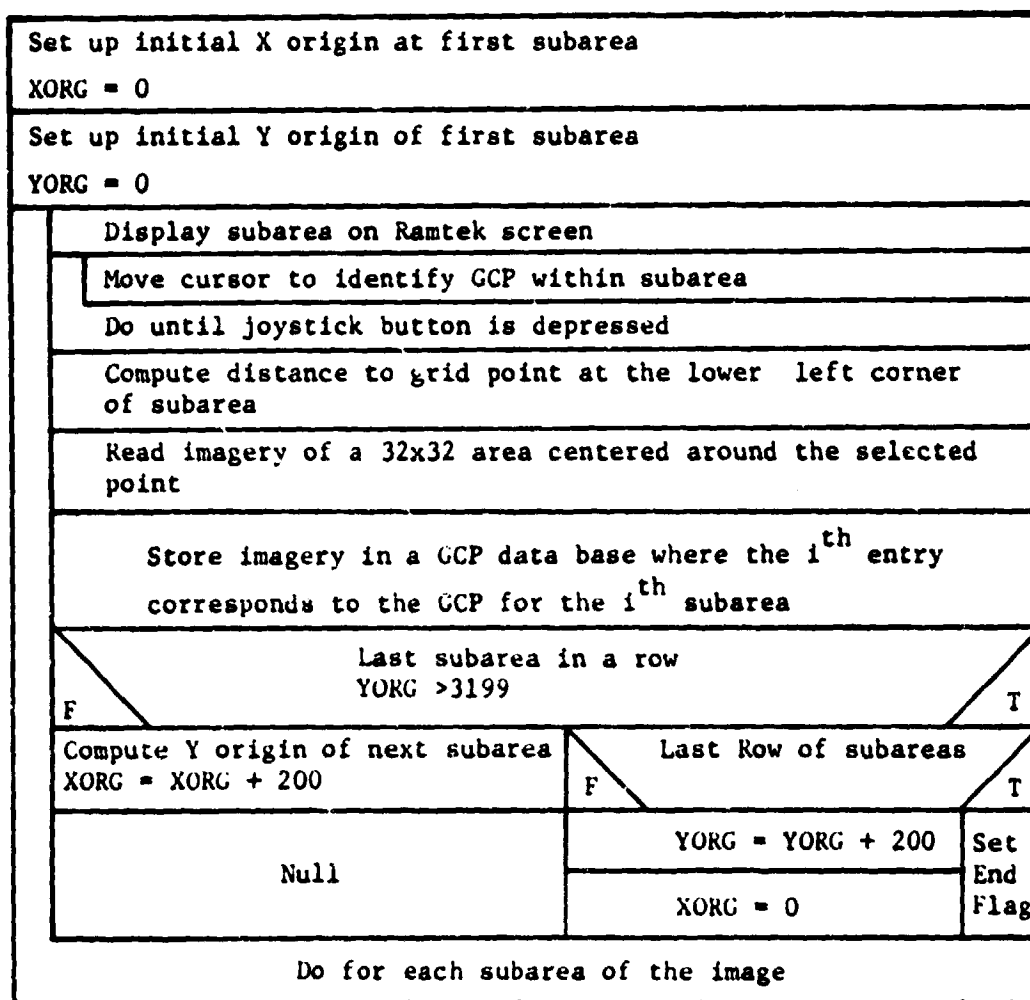


Figure V-2 GCP Selection

The correlation analysis program has also been modified to automatically register the 400 GCPs from the reference scene on the corresponding image being studied. For each GCP detection that does not encounter a false lock, a registration vector depicting the relative offsets of the two images will be displayed with the base of the vector originating from the center of the GCP location in the reference grid (Fig. V-3). A VCLR of this process is shown in Figure V-4.

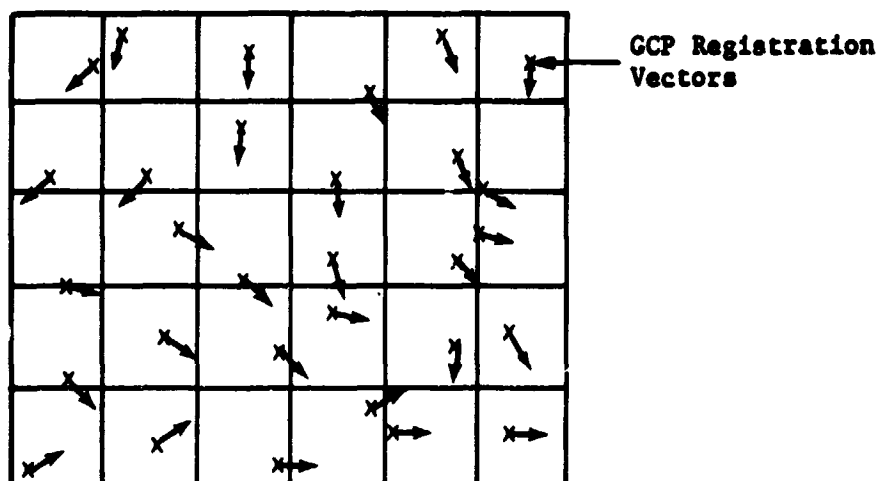


Figure V-3
Registration Vector Display

A VCLR of this process is shown in Figure V-4.

Search area center = GCP location	
Search area size = 52	
Read in search area from second image	
Read in error data base	
Perform SSDA correlation and compute false lock coefficient	
False lock coefficient > 0.9	
F	T
DISX = 999.0	Compute local distortion vector
DISY = 999.0	
DISX = REGX - 10	
DISY = REGY - 10	
Store local distortion vector	Store local distortion vector
Do for each GCP	
Print distortion vectors	
Plot distortion vectors	

Figure V-4 Distortion Measurement Program

It is interesting to note that since CORALA provides a measure of the registration offset at a single point, distortion measurement may be performed without any program modifications. However, the entire process of computing the distortions at each grid point and the distortion vectors must be done by hand.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

One of the key bottlenecks associated with NASA's end-to-end data management problem is the process of image correction. Resampling not only requires a tremendous amount of data processing but also requires the integration of volumes of data from every aspect of the mission, including gyro data, tracking data, and imagery.

With the development of the MLA, autonomous real-time image correction will become feasible. The distortions associated with the multi-spectral scanner's nonlinear mirror sweep and nonuniform sweep period will be eliminated, leaving a well-defined distortion caused by earth curvature and look angle. The tremendous increases in computational capabilities both on the ground and aboard the spacecraft will also simplify the problem.

A solution to the problem may be realized through either ground processing or spaceborne processing. However, even though ground-based processing may alleviate the immediate problem, in the long range adaptive systems using onboard intelligence will be required. Therefore it may be advantageous to begin the transition to onboard automated systems now.

Many benefits may be realized through onboard spacecraft control. By isolating the science sensor from the attitude limit cycle of the vehicle using a pointing mount, a curved focal plane MLA that provides uniform sampling of the image can be incorporated, thus reducing the need for along-scan resampling. This approach, although attractive for missions such as Landsat where the revisit cycle is periodic, may not be attractive for missions requiring deterministic data acquisition through sensor pointing because a curved focal plane array boresight must always point at nadir. However, isolation of the sensor from the attitude limit cycle will simplify the image correction process even without the curved focal plane array.

One of the keys to future remote sensing missions is the onboard determination of sensor boresight position in earth-fixed coordinates. Both image correction and deterministic data acquisition systems will require this knowledge to be useful. Analysis has shown that inertial systems alone do not provide sufficient accuracy for several reasons:

- 1) It is difficult to resolve misalignments between inertial sensor frames and science sensor frames;
- 2) Misalignment between inertial and local earth frames are difficult to compute;
- 3) Star trackers are not accurate enough.

Ground control point (GCP) sightings can be used to directly solve for these misalignments and hence reduce the error in boresight determination.

Analysis has shown that a registration accuracy of 0.1 pixels can be achieved using an SSDA correlation algorithm. It is also possible to detect situations where the correlator locks onto a false target. These factors make the correlator an attractive navigation sensor for an onboard image correction system.

A system comprising two NASA standard star trackers, a NASA standard gyro package, a GPS receiver, and the landmark tracker with a 30-meter resolution was simulated. Analysis results indicate that with a GPS update frequency of between 5 and 10 seconds and a GCP sighting frequency of between 20 and 80 seconds, the position of the sensor boresight can be determined to within 15 meters. With the one exception of the MLA, the entire system can be implemented with existing technology. The most restricting feature of the system is the tremendous storage requirements for the GCP data base. Assuming that GCPs may only be located on land (accounting for 20% of the earth's surface) and assuming a GCP update frequency every 40 seconds or approximately every 1 1/3 scenes, the total storage requirements may be found by:

$$\begin{aligned}\text{Total earth surface area} &= 4\pi(6 \times 10^3)^2 \text{ km}^2 \\ &\approx 4.5 \times 10^8,\end{aligned}$$

$$\begin{aligned}\text{Total number of available scenes} &\approx 4.5 \times 10^8 \text{ km}^2 \times 0.2 \div 175^2 \text{ km}^2 \\ &= 2.9 \times 10^3,\end{aligned}$$

$$\begin{aligned}\text{Total number of GCPs} &= 2.9 \times 10^3 \times 3/4 \\ &\approx 2 \times 10^3,\end{aligned}$$

$$\text{Total memory required} = 2 \times 10^3 \text{ GCPs} \times 32^2 \frac{\text{pixels}}{\text{GCP}} \times \frac{8 \text{ bits}}{\text{pixel}} = 18 \times 10^6 \text{ bits}.$$

Although this is a tremendous storage requirement, it is still within the limits of current technology. This volume may also be reduced by a factor of two by reducing the number of gray scales in the GCP reference from 256 to 16. If histogram equalization is used to reduce the number of gray scales, the accuracy of the correlation may not be affected. This possibility should be investigated.

It is estimated that processing requirements can be met with the use of two general-purpose flight computers and a hardwired correlator. The separation of functions would be directly analogous to the navigation and the registration processors discussed in Chapter II.

B. RECOMMENDATIONS

If the landmark tracker is to be realized as an operational system, a well-planned development schedule is required. The critical tasks that must be performed include:

- 1) Further analysis of the correlator is required to determine its sensitivity to seasonal variations. Several passes over the same scene are required and the GCP analysis process must be repeated for each pass. It is also necessary to determine the types of landmarks that produce the best correlation. This information will be invaluable in the selection of GCPs for an operational system;
- 2) Research must be conducted to determine methods of reducing the storage requirements of the GCP data base. The most promising approach seems to be a mixture of image enhancement and gray scale reduction;
- 3) A study must be conducted to establish the configuration of the on-board processing network;
- 4) The entire system should be breadboarded and ground-tested to resolve peculiarities of the design;
- 5) The breadboard system may be flown on a low-cost aircraft flight test to more fully demonstrate feasibility;
- 6) The breadboard system could also be incorporated into the IAS ground demonstration as an additional step toward a total system configuration.

The final development stage of the system would be incorporation of the design into an advanced remote sensing mission such as OERS.

The curved focal plane MLA is another concept that should be investigated. If feasible, the concept would dramatically reduce the processing requirements associated with remote sensing missions. This investigation can be separated into two tasks. First, the feasibility of the concept must be analyzed. Technical considerations include:

- 1) Precision required in machining of the curved surface;
- 2) Effect of variations in local earth curvature on registration accuracy;
- 3) Pointing requirements needed to ensure 1/2 pixel registration.

The second task that must be performed is a cost/benefit tradeoff between deterministic data acquisition through sensor pointing versus image correction using the curved focal plane array. The two approaches are mutually exclusive.

VII. REFERENCES

1. Space Systems Technology Model. National Aeronautics and Space Administration Office of Aeronautics and Space Technology, Code RS, NASA Headquarters, Washington, DC, May 1980.
2. P. C. Carney et al.: Onboard Attitude Determination System Final Report, NAS4-23428 Mod. 27, Martin Marietta Denver Aerospace, 1978.
3. H. M. Thomas et al.: Definition of Second Feature Identification and Location Experiment (FILE-II). MCR-81-539, Prepared under Contract NAS1-15737, March 1981.
4. R. L. White et al.: "Attitude and Orbit Estimation Using Stars and Landmarks." IEEE Transactions on Aerospace and Electronic Systems, Vol AES-11, No. 2, 1974.
5. R. L. White et al.: "Use of Known Landmarks for Satellite Navigation." AIAA Paper No. 75-1097, 1975.
6. N. F. Toda and F. H. Schlee: "Autonomous Orbital Navigation by Optical Tracking of Unknown Landmarks." Journal of Spacecraft and Rockets, Vol 4, No. 12, 1967.
7. D. H. Aldrich: "Interferometer Landmark Tracker Navigation System." Neacon Journal, 1974.
8. D. I. Barnea and H. F. Silverman: "A Class of Algorithms for Fast Digital Image Registration." IEEE Transactions on Computers, Vol C-21, No. 2, February, 1972.
9. R. T. Schappell et al.: Experimental and Simulation Study Results for Video Landmark Acquisition and Tracking Technology. NASA Contractor Report 158997 prepared under Contract NAS1-14489, February 1979.
10. L. E. Gilbert et al.: "Onboard Landmark Navigation and Attitude Reference Parallel Processor System." Paper presented at the Second Annual Flight Mechanics and Estimation Theory Symposium, 1977.